



COLOR COMPUTER



MC1000

MANUAL DE INSTRUÇÕES E BASIC

MC1000

MANUAL DE INSTRUÇÕES E BASIC

1ª EDIÇÃO

MANUAL DE INSTRUÇÕES E BASIC

MC-1000

SUMARIO

CAPITULO 1

INTRODUÇÃO	8
O QUE VOCÊ PODE FAZER COM O SEU MICROCOMPUTADOR	8
O QUE É UM PROGRAMA	8
O QUE É A LINGUAGEM BASIC	8

CAPITULO 2

INSTALANDO O SEU MC-1000	9
A EMBALAGEM	9
O QUE VOCÊ PRECISARÁ PARA INSTALAR O SEU MC-1000	9
LIGANDO SEU MICRO A TV	9
O TECLADO	11
ALGUMAS TECLAS ESPECIAIS	11
INSTRUÇÕES CONJUGADAS AS TECLAS	12
PRECAUÇÕES	13

CAPITULO 3

EXPERIMENTANDO ALGUMAS CARACTERÍSTICAS ÚNICAS DO SEU MC-1000.	15
SOM	15
COR	16

CAPÍTULO 4

INFORMAÇÕES ÚTEIS	17
A TECLA RETURN	17
EXECUÇÃO DIRETA E PROGRAMADA	17
OPERADORES NUMÉRICOS	18
PRIORIDADE DE EXECUÇÃO	18
NÚMEROS INTEIROS E REAIS	18
PALAVRAS RESERVADAS	19
VARIÁVEIS E CONSTANTES NUMÉRICAS	20
VARIÁVEIS E CONSTANTES ALFANUMÉRICAS	20

CAPÍTULO 5

PRONTO PARA PROGRAMAR	22
INSTRUÇÃO LET	22
INSTRUÇÃO PRINT	22
LIMPANDO A TELA	23
O COMANDO RUN	23
COMEÇANDO UM NOVO PROGRAMA	24
A INSTRUÇÃO INPUT	25
LENDO DADOS	25
PARA LISTAR UM PROGRAMA	27
NUMERANDO AUTOMATICAMENTE AS LINHAS	27
COMENTÁRIOS EM UM PROGRAMA	28
OS COMANDOS END E STOP	28
A INSTRUÇÃO CLEAR	29

CAPÍTULO 6

EDITANDO SEU PROGRAMA	30
INFORMAÇÕES ADICIONAIS SOBRE O COMANDO LIST	30
APAGAR PARTE DA LINHA CORRENTE	30
INSERIR UMA NOVA LINHA	31
CORRIGIR UMA LINHA JÁ EXISTENTE	31
A FUNÇÃO L	32
A FUNÇÃO I	32
A FUNÇÃO D	33
A FUNÇÃO S	33
A FUNÇÃO H	34
A FUNÇÃO X	34
A FUNÇÃO Q	35

CAPÍTULO 7

LOOPS, DESVIOS E DECISÕES	36
FOR...TO...STEP...NEXT	36
GOTO	37
IF...GOTO	38
IF...THEN	38
ON...GOTO	39
OPERADORES RELACIONAIS.....	39
OPERADORES LÓGICOS (BOOLEANOS)	40
ORDEM COMPLETA DE PRECEDÊNCIA	41

CAPÍTULO 8	
SUB-ROTINAS	42
O QUE É UMA SUB-ROTINA	42
AS INSTRUÇÕES GOSUB E RETURN	42
SUB-ROTINA CONDICIONAL MÚLTIPLA	43
CAPÍTULO 9	
FUNÇÕES NUMERICAS	44
FUNÇÕES TRIGONOMETRICAS	44
ALGUMAS FUNÇÕES MATEMATICAS	44
A FUNÇÃO RND	45
A INSTRUÇÃO DEF FN	45
CAPÍTULO 10	
SOM E MÚSICA	46
AS INSTRUÇÕES TEMPO E SOUND	46
TABELA DE CÓDIGO PARA NOTAS MUSICAIS	47
CAPÍTULO 11	
GRÁFICOS E CORES	48
CORES	48
TEXTOS E GRÁFICOS	48
AS INSTRUÇÕES PLOT E UNPLOT	49
AS INSTRUÇÕES DRAW E UNDRAW	51
TROCA DE ORIGEM	51
DISPLAY NORMAL E INVERSE	52
A INSTRUÇÃO COLUMN	52

CAPÍTULO 12

STRINGS E MATRIZES	53
A FUNÇÃO ASC	53
A FUNÇÃO CHR\$	53
CONCATENAÇÃO DE STRINGS	54
A FUNÇÃO LEN	54
AS FUNÇÕES LEFT\$, RIGHT\$ E MID\$	55
A FUNÇÃO STR\$	56
A FUNÇÃO VAL(\$\$)	56
MATRIZES NUMÉRICAS E ALFANUMÉRICAS	56
AS DIMENSÕES DE UMA MATRIZ	58
MATRIZES DE ORDEM SUPERIOR	59

CAPÍTULO 13

ARMAZENANDO E CARREGANDO PROGRAMAS EM FITA CASSETE	60
O GRAVADOR CASSETE	60
OS COMANDOS SAVE E LOAD	61
PARA GUARDAR E CARREGAR DADOS	61

CAPÍTULO 14

INSTRUÇÕES RELACIONADAS AO SISTEMA	64
A INSTRUÇÃO FRE	64
MAPA DE MEMÓRIA	64
AS INSTRUÇÕES PEEK E POKE	66
O CURSOR	66
PORTAS DE ENTRADA E SAÍDA (E/S)	67

A INSTRUÇÃO OUT	67
A INSTRUÇÃO INF	67
A INSTRUÇÃO WAIT	67

CAPÍTULO 15

OUTROS COMANDOS E INSTRUÇÕES	69
O COMANDO PR#	69
PARA DEPURAR UM PROGRAMA	69
TABULAÇÃO	70
A INSTRUÇÃO WIDTH	70

CAPÍTULO 16

DESENVOLVIMENTO DE SOFTWARE E DEPURAÇÃO (DEBUGGING) EM LINGUAGEM DE MÁQUINA	72
O COMANDO DEBUG	72
O SUBCOMANDO D	73
O SUBCOMANDO F	73
O SUBCOMANDO S	74
O SUBCOMANDO M	74
O SUBCOMANDO G	75
O SUBCOMANDO X	75
O SUBCOMANDO R	76
FINALIZAR UMA SUB-ROTINA EM LINGUAGEM DE MÁQUINA	76
RETORNAR AO BASIC	76
CHAMAR UMA SUB-ROTINA EM LINGUAGEM DE MÁQUINA	77

CAPÍTULO 17

PROGRAMAS DA FITA CASSETE	78
INSTRUÇÕES	79
MARS LANDING	80
CITY BOMBER	81
KINDERGARTEN-MATH	82
TANK-BATTLE	83
MUSIC COMPOSER	84
OS TRÊS CANAIS DA MÚSICA	84
PROCEDIMENTO PARA A COMPOSIÇÃO	85

APÊNDICES

APÊNDICE A: MENSAGENS DE ERRO	91
APÊNDICE B: PALAVRAS RESERVADAS	94
APÊNDICE C: ACESSÓRIOS OPCIONAIS	95

CAPÍTULO 1

INTRODUÇÃO

O QUE VOCÊ PODE FAZER COM SEU MICROCOMPUTADOR

Uma aplicação imediata para seu micro é aprender a programar na linguagem BASIC e desenvolver programas para facilitar seu trabalho ou utilizar programas que podem ajudá-lo de muitas maneiras: para cuidar de suas despesas pessoais, para fins educacionais ou apenas com seus video games.

Tente pensar em mais coisas que você gostaria que ele fizesse para você.

O QUE É UM PROGRAMA

Um programa de computador é simplesmente uma sequência de instruções organizadas logicamente com uma finalidade pré-definida. Ele diz ao computador o que você quer que ele faça.

O QUE É A LINGUAGEM BASIC

A palavra BASIC aqui significa "Beginners All-purpose Symbolic Instruction Code". A linguagem BASIC atualmente é uma das linguagens mais populares no mundo todo, e vem sendo utilizada em quase todo tipo de computadores.

CAPÍTULO 2

INSTALANDO O SEU MC-1000

A EMBALAGEM

Quando você abrir a embalagem do seu MC-1000, você encontrará os seguintes equipamentos:

- 1- o MC-1000 COLOR COMPUTER;
- 2- este manual e o manual de referência;
- 3- um balun para TV (a caixinha preta, cujo cabo liga o micro a antena da TV);
- 4- uma fonte de alimentação (a caixa com os pinos para ligar na tomada);
- 5- dois cabos para ligar o micro ao gravador cassete;
- 6- uma fita cassete com programas.

O QUE VOCÊ PRECISARÁ PARA INSTALAR SEU MC-1000

Para você instalar o seu microcomputador da forma mais eficiente, você precisará:

- um televisor colorido ou um monitor (o televisor deve funcionar no sistema PAL-M);
- um gravador cassete;
- uma chave de fenda para soltar os parafusos da antena na parte de trás de sua televisão.

LIGANDO SEU MICRO A TV

Atenção: verifique a voltagem da rede local (110 ou 220V) e faça o ajuste, se necessário, em sua fonte de alimentação. A chave seletora de voltagem encontra-se em sua parte superior, junto as indicações do fabricante.

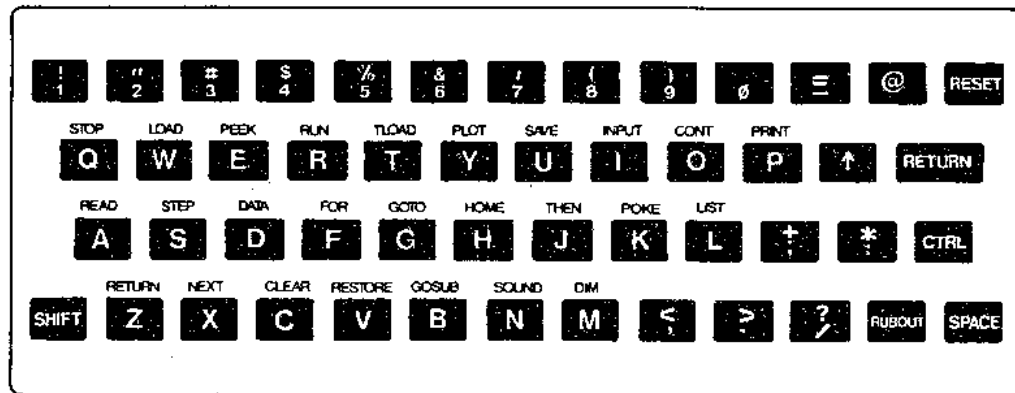
Oriente-se na figura abaixo que mostra os vários conectores da parte traseira do micro.

Ligue o plug da ponta do cabo coaxial que se adapta ao conector marcado "RF" e a outra ponta a entrada da antena da TV. Afrouxe os parafusos da antena de VHF na parte de trás da TV, coloque os dois pequenos terminais na antena e aperte os parafusos. Se a sua TV usa tomada ao invés de terminal de antena, peça ajuda ao seu revendedor.

Após ter acoplado seu micro a TV, ligue e ajuste-a no canal 13. Agora ligue a fonte de alimentação na tomada e o plug da fonte no conector indicado com "9V DC". A tela ficará como na foto abaixo, sendo que o número de bytes dependerá do uso ou não de expansões. Após isso, aparecerão na tela o sinal "OK" e o cursor (retângulo piscante).

O TECLADO

A figura abaixo mostra o teclado do MC-1000. Você vai notar que o desenho do teclado é muito parecido ao da máquina de escrever. Você pode escrever letras, números e outros caracteres. Pressione a tecla <SHIFT> quando quiser usar os caracteres impressos na parte superior das teclas.



ALGUMAS TECLAS ESPECIAIS

A tecla <SHIFT> é usada para se usar o caracter da parte superior de uma tecla, caso a mesma possua 2 caracteres. Quando combinada as teclas de <A> até <Z>, ela indica o uso das funções pré-determinadas correspondentes.

Quando a tecla <RESET> é digitada em conjunto com a tecla <SHIFT>, faz com que a memória do micro volte ao seu estado inicial, isto é, os ponteiros são inicializados. Para usar as teclas conjugadas com <SHIFT> faça o seguinte: primeiro pressione a tecla <SHIFT> e mantenha-a assim e, então digite a tecla que desejar. Após ter feito isso, solte as duas teclas.

NOTA: se você não puder dar um reset no computador pelo <SHIFT>, você pode desligá-lo e ligá-lo novamente que terá o mesmo efeito.

A tecla <RETURN> deve ser usada após cada entrada, seja ela uma linha de comando ou a resposta a um INPUT. Esta tecla serve para indicar ao computador que aquilo que foi digitado já pode ser processado.

A tecla <RUBOUT> faz com que o cursor retorne uma posição cancelando o caracter por sobre o qual passa.

A tecla <CTRL> é uma tecla de controle. Esta tecla é usada em combinação com outras para executar certas funções. Ex: CTRL-H executa a função de retorno do cursor numa mesma linha, caracter por caracter (faz a função do <RUBOUT>).

NOTA: será usada a seguinte notação para indicar a combinação da tecla <CTRL> com uma outra. Por exemplo, ^<K> ou simplesmente ^K significa que a tecla <CTRL> deve ser pressionada juntamente com <K>.

INSTRUÇÕES CONJUGADAS AS TECLAS

Para conveniência e rapidez, o teclado do MC-1000 foi preparado para que os 26 comandos BASIC mais usados possam ser escritos apenas usando uma tecla de letra conjugada com <SHIFT>. Aqui temos a lista das teclas e seus comandos correspondentes:

TECLA	INSTRUÇÃO	TECLA	INSTRUÇÃO
A	READ	N	SOUND
B	GOSUB	O	CONT
C	CLEAR	P	PRINT
D	DATA	Q	STOP
E	PEEK	R	RUN
F	FOR	S	STEP
G	GOTO	T	TLOAD
H	HOME	U	SAVE
I	INPUT	V	RESTORE
J	THEN	W	LOAD
K	POKE	X	NEXT
L	LIST	Y	PLOT
M	DIM	Z	RETURN
N	SOUND		

Para que esta facilidade fosse possível, foi necessário suprimir as letras minúsculas, portanto só é possível o uso de letras maiúsculas.

Ao pressionar qualquer tecla, um bip será emitido indicando o ocorrido. Este som não será gerado se você estiver utilizando programas específicos que o suprimem ou jogos.

PRECAUÇÕES

Abaixo estão relacionados os cuidados que você deve ter com seu MC-1000.

- 1- Sempre segure-o com cuidado, evitando impactos fortes.
- 2- Conserve-o longe de líquidos.
- 3- Guarde-o num lugar seco e evite altas temperaturas sobre ele.
- 4- Desligue a alimentação e tire a fonte da tomada quando o mesmo não estiver em uso.
- 5- Certifique-se de que ele está desligado antes de conectar qualquer dispositivo externo, caso contrário, o dispositivo e o micro serão danificados.
- 6- Periodicamente, limpe-o com um pano umedecido com água e sabão. Tome cuidado para que não vá água para o interior do gabinete. Nunca use álcool ou outro solvente.

Caso ocorram problemas com seu micro, confira a instalação, conexões e verifique os itens relacionados abaixo:

1- NENHUM SINAL NA TELA.

- A TV pode não estar bem sintonizada ou no canal errado.
- Mau contato no conector ou no cabo que liga o micro à TV.
- Ligação deste cabo em posição errada (plug, TV ou monitor).

2- FUNCIONAMENTO ANORMAL

- O micro pode apresentar caracteres estranhos ou não funcionar corretamente. Desligue-o por uns instantes e ligue-o novamente.
- Caracteres não pressionados surgem na tela; mesma solução do item anterior.
- Utilização errada nas combinações de teclas do tipo <CTRL> ou <SHIFT>.

3- O COMPUTADOR TRAVA, PARA A EXECUÇÃO OU NÃO ACEITA NOVOS COMANDOS.

- O computador é instruído para enviar dados ou obter dados dos dispositivos externos (tais como impressora, gravador ou drive) quando na realidade nenhum dispositivo está conectado.
- O computador é instruído para executar uma sub-rotina em linguagem de máquina que não tem um endereço de final ou não tem instruções para retornar o controle para o BASIC.
- Solução: desligue e ligue o computador novamente ou pressione <CTRL> <RESET>.

4- ENTRADA OU SAÍDA DE CASSETE NÃO FUNCIONA

- Mau contato.
- O volume do gravador cassete não se encontra no nível correto.
- Nome do arquivo errado.
- Uso incorreto da instrução.
- Gravador danificado.

CAPÍTULO 3

EXPERIMENTANDO ALGUMAS CARACTERÍSTICAS INTERESSANTES E ÚNICAS DO SEU MC-1000

Este capítulo destina-se àqueles que querem ver alguma coisa antes de ler todo este manual. Nós daremos alguns exemplos de programas para demonstrar algumas características interessantes do MC-1000. Mas não explicaremos o programa aqui, você aprenderá como fazer programas similares depois.

Agora ligue o computador e digite os seguintes programas um de cada vez.

NOTA: após você ter digitado uma linha certifique-se que não há nenhum erro, só então tecle <RETURN> e passe para a próxima linha. Se existir um erro de escrita, tecle <RUBOUT> para apagar os caracteres errados, então digite os caracteres corretos e tecle <RETURN>. Consulte o capítulo 6 se encontrar alguma dificuldade.

SOM

Digite:

```
10 TEMPO 150,2,1
20 SOUND 87,15,1
30 SOUND 96,15,1
40 GOTO 20
RUN
```

Agora ajuste o som da TV no volume que preferir. A TV fará o som de uma sirene de ambulância. Para pará-lo, pressione as teclas <CTRL> e <C> simultaneamente.

COR

Agora digite:

```
NEW
10 HGR
20 FOR A = 1 TO 10
30 B = B + 10
40 PLOT 30+B,60 TO 30+B,180 TO 90+B,120 TO 30+B,60
50 NEXT
60 FOR I = 1 TO 500:NEXT
RUN
```

Para voltar ao modo texto, digite TEXT sem esquecer do <RETURN>.

O exemplo apresenta o modo gráfico de Alta Resolução (HGR) que, como você pode comprovar é preto e branco. Para obter cores, você terá que usar o modo gráfico de Baixa Resolução (GR), que possui oito cores. Tente modificar o programa acima para o modo gráfico de Baixa Resolução e utilize a instrução COLOR, e acontecerá algo mais interessante.

O formato do comando COLOR será introduzido mais tarde.

CAPÍTULO 4

INFORMAÇÕES ÚTEIS

A TECLA <RETURN>

Pressione a tecla <RETURN> uma vez. Isto diz ao computador que você terminou de fazer uma entrada. Caso você não o faça, o computador irá tratar todos os caracteres subsequentes como parte da entrada corrente. Sendo assim, teclé sempre <RETURN> após cada linha de programa, instrução ou resposta a INPUT. Portanto, consideraremos que você já sabe isto e não será necessário mostrá-lo a cada exemplo.

EXECUÇÃO DIRETA E PROGRAMADA

Seu computador pode executar instruções em ambos os modos, direto ou programado. No modo direto, você simplesmente digita a instrução e tecla <RETURN>. Por exemplo:

```
X=6+2+7:PRINT X
```

O computador faz o cálculo e exibe o número 15 na tela. Depois da execução da instrução, o valor de X é armazenado na memória, mas a instrução será esquecida.

No modo programado você tem que determinar um número para cada linha de instrução. Uma linha pode consistir de uma ou mais instruções. Estas linhas serão guardadas na memória para posterior uso. Depois que você introduzir todas as linhas de instrução, você pode executar (RUN) o programa quantas vezes quiser. Por exemplo:

```
10 PRINT 3+4+5+2  
20 PRINT 3+9-5  
30 PRINT 45294-13912  
RUN
```

As instruções com número de linha 10, 20 e 30 são guardadas na memória. Somente depois que você digitar RUN e pressionar <RETURN> a tela mostrará 14, 8 e 31382. Note que a instrução RUN deve ser executada no modo direto. Muitas instruções podem ser executadas em ambos os modos, mas algumas só podem ser executadas em um dos dois modos.

Os números de linha devem ser inteiros e estar dentro de 0 a 65535. Você pode introduzir programas sem obedecer a ordem crescente de números, mas o computador irá executá-las na sequência, do menor para o maior número de linha. Sugerimos usar números de linha em espaços de 10 para que, mais tarde, você possa acrescentar instruções entre duas já existentes, se necessário.

OPERADORES NUMÉRICOS

Os seguintes operadores numéricos são usados em BASIC:

- + adição
- subtração
- * multiplicação
- / divisão
- negativo de um número
- = igualar duas expressões ou dar um valor a uma variável
- ^ exponenciação

PRIORIDADE DE EXECUÇÃO

Abaixo mostramos a ordem que o computador executa as operações:

- 1º: () expressões dentro de parênteses
- 2º: ^ exponenciação
- 3º: - negação
- 4º: * / multiplicação e divisão
- 5º: + - adição e subtração

Quando duas ou mais operações em uma expressão tem a mesma prioridade, o computador as faz da esquerda para a direita.

NÚMEROS INTEIROS E REAIS

Números inteiros são todos os números incluindo o zero, podem ser positivos ou negativos e não tem parte fracionária.

Números reais são todos os números positivos e negativos incluindo os números decimais e inteiros. Por exemplo, -1,45 e 67 são números reais.

Seu computador toma todos os números reais como decimais com seis dígitos significativos após a vírgula. Por exemplo:

7 é tratado como 7.00000
-2 é tratado como -2.00000
3.5 é tratado como 3.50000
4.836757 é tratado como 4.83676

Um número com mais de 6 dígitos significativos será arredondado para 6 dígitos. A notação de ponto flutuante (ou chamada notação científica) é usada para mostrar números com mais de seis dígitos colocados à direita ou à esquerda da vírgula. Por exemplo:

12.3789000 é mostrado como 1.23789E+08
(tente mover a vírgula 8 espaços para a esquerda)

0.0002468 é mostrado como 2.468E-04
(tente mover a vírgula 4 espaços para a direita)

-0.00000000037 é mostrado como -3.7E-10

Aqui 1.23789E+08 significa 1.2389×10^8
-3.7E-10 significa -3.7×10^{-10}

E+08 representa a parte exponencial da notação científica.

Você não deve preocupar-se com esta notação caso não esteja familiarizado com ela, pois ela não será usada por hora.

Seu micro pode trabalhar com números reais de $1E+38$ a $-1E+38$. Os números mais próximos de zero são: pelo lado positivo $1E-38$ e $-1E-38$ pelo lado negativo.

Uma vez que o computador arredonda os números para 6 dígitos significativos com os limites acima, o resultado de alguns cálculos poderão ser levemente diferentes do que você espera. Você pode arranjar a aritmética de maneira diferente caso o computador lhe mande uma mensagem de erro SE (sem espaço).

PALAVRAS RESERVADAS

Algumas palavras ou combinações de letras são reservadas para o BASIC. Elas informam ao computador o que deve ser feito. Veja no apêndice B a relação das palavras reservadas.

VARIÁVEIS E CONSTANTES NUMÉRICAS

Constantes numéricas são números com valores fixos, por exemplo, 841 e -31. Variáveis numéricas são quantidades matemáticas às quais podem ser atribuídos diferentes valores numéricos. Os valores não são fixados até que eles sejam relacionados às variáveis. Você deve dar nomes às variáveis para que o computador determine áreas na memória para as mesmas.

Nomes para variáveis numéricas podem ter qualquer combinação de letras e números, observando-se que:

- nenhuma palavra reservada apareça no nome;
- o primeiro caracter do nome seja uma letra;
- o comprimento do nome não exceda uma linha de programa.

Note que o computador somente reconhece os 2 primeiros caracteres do nome, não importando que o mesmo tenha mais de dois caracteres. Ambas as variáveis ETA e ETO são tratadas como iguais.

VARIÁVEIS E CONSTANTES ALFANUMÉRICAS

Seu computador é mais poderoso do que uma calculadora. Ele trabalha com textos tal como com números. Uma cadeia ou sequência de caracteres é qualquer combinação de letras, números e caracteres colocados dentro de aspas. O nome que adotaremos daqui para frente é "string", pois nem cadeia nem sequência de caracteres são definições corretas.

O computador usa strings para trabalhar com textos. Exemplos:

"45735"

"MC-1000 CCE" (os espaços dentro das aspas fazem parte do string)

"6B2E"

""

(um par de aspas sem nenhum caracter e nem espaços, é chamado STRING NULO)

Os strings nos exemplos acima são chamados constantes porque eles fixaram "valor", ou seja, fixaram caracteres. Você pode também dar qualquer nome a um string observando:

- o nome não contém nenhuma palavra reservada;
- o primeiro caracter no nome seja uma letra;
- o nome não exceda o comprimento de uma linha de instrução;
- o último caracter é um símbolo do dólar (\$);

Com o sinal de dólar (\$), o computador sabe que ele está trabalhando com uma variável string.

Exemplos:

NOME DE VARIÁVEIS NUMERICAS

TOT1	(OK)
1TOT	(erro, início com número)
COLORIDO	(erro, palavra reservada)
CALORIDO	(OK)

NOMES DE VARIÁVEIS TIPO STRING

TOT1\$	(OK)
1TOT\$	(erro, início com número)
PAG	(erro, falta "\$")
PRINT\$	(erro, palavra reservada)

CAPÍTULO 5

PRONTO PARA PROGRAMAR

Agora você poderá começar a fazer pequenos programas. Siga todos os exemplos, à medida que forem aparecendo. Digite seus programas e confira-os cuidadosamente antes de executá-los. Se você cometer erros de digitação, antes de pressionar <RETURN> use <RUBOUT> para apagá-los. Se você mais tarde achar um erro, reescreva a linha de forma correta. No capítulo 6 serão mostrados vários métodos de correção.

Em qualquer caso, aparecerá na tela uma mensagem de erro mostrando a linha em que o erro ocorreu e o código correspondente.

A INSTRUÇÃO LET

A instrução LET é usada para atribuir um valor definido a uma variável. Exemplo:

```
LET A = 4+3+84  
LET BC = 78
```

A palavra LET é opcional, você pode simplesmente digitar:

```
A = 4+3+84  
BC = 78
```

A INSTRUÇÃO PRINT

Você pode usar a instrução PRINT para exibir mensagens, números, símbolos ou resultados de cálculos na tela. Exemplo:

```
A = 13+4+7 : PRINT A
```

O resultado apresentado na tela é 24. Note que neste caso, você está usando o modo direto. Note também que existem duas instruções na mesma linha.

Cada linha de programa poderá ter um comprimento máximo de 255 caracteres, incluindo espaços. Você pode usar mais de uma instrução em uma linha de programa, sendo que cada instrução deve ser separada por dois-pontos (:).

Faça o seguinte:

```
AREA = 3*3
PRINT "A AREA DO MEU QUARTO E ";AREA;" METROS
QUADRADOS"
```

NOTA: se você quiser exibir mais de uma mensagem na mesma linha, você deve separá-las com ponto e vírgula (;).

LIMPANDO A TELA

Algumas vezes é desejável limpar a tela e começar colocando instruções desde o topo da mesma. Digite HOME e pressione <RETURN>. A tela é limpa e o cursor é levado ao canto superior esquerdo da tela. Você pode também usar a instrução HOME no modo programado. Agora tente isto:

```
HOME
PRINT "SUZANE"
```

O COMANDO RUN

Digite o seguinte programa:

```
10 A=1000
20 PRINT "OLA! EU SOU O MC";A
30 PRINT "COLOR COMPUTER"
40 PRINT "DA CCE"
```

Agora digite RUN e pressione <RETURN>. O comando RUN instrui o computador ou parte dele, iniciando pelo número de linha dado. Exemplo:

```
RUN 30
```

Outro exemplo:

```
10 PRINT "AQUI ESTA OUTRO PROGRAMA"
20 L = 25
30 PRINT "O COMPRIMENTO DE UM LADO DE UM QUADRADO E
";L;" METROS"
40 PRINT "O PERIMETRO E ";4*L;"METROS"
50 PRINT "A AREA DO QUADRADO E "; L^2; "METROS
QUADRADOS"
RUN
```

COMEÇANDO UM NOVO PROGRAMA

Quando você quiser digitar um novo programa, é recomendado apagar todo o programa e dados anteriormente digitados, a não ser que você deseje mantê-los na memória. Neste caso, tenha cuidado em usar números de linha diferentes dos já usados. Digite o seguinte:

```
10 PRINT "ESTE E UM PROGRAMA VELHO"
20 A= 3+3+5+4 : B= 9+2+5+7
30 PRINT A+B;A*B
40 PRINT (A*B)/(A+B)
RUN
```

Digite:

```
NEW
15 PRINT "ESTE E UM NOVO PROGRAMA"
25 A = 7+7+9;B=17+16+19
35 PRINT A;" E UM NUMERO IMPAR"
45 PRINT "O RESULTADO DE A+B E IGUAL A "; A+B
RUN
```

Como você pode ver, o programa velho foi apagado e só ficou o novo na memória.

A INSTRUÇÃO INPUT

Algumas vezes é desejável parar a execução de um programa e adicionar dados ao mesmo. A instrução INPUT exhibe um ponto de interrogação e espera que você digite estes dados. Exemplo:

```
10 INPUT "QUAL A SUA IDADE?";A
20 PRINT "SUA IDADE E"; A
30 INPUT "A IDADE DO SEU IRMAO E? ";B
40 PRINT "A SOMA DE SUAS IDADES E ";A+B;" ANOS"
RUN
```

Quando você vê o ponto de interrogação, digite a resposta e pressione <RETURN>. Agora veja outro exemplo:

```
10 INPUT A,B,C
20 PRINT "A SOMA DE";A;B;C;"E IGUAL A "; A+B+C
```

NOTA: se você colocar mais de uma variável depois de INPUT certifique-se de que elas estejam separadas por vírgula. Os dados também devem ser sepados por vírgula.

Agora veja um exemplo com a variável string:

```
10 INPUT "QUAL E SEU NOME "; A$
20 INPUT "QUAL O NOME DO SEU IRMAO "; B$
30 PRINT "VAMOS VER OS DOIS NOMES JUNTOS "
40 PRINT
50 PRINT A$;B$
```

LENDO DADOS

Ao invés de usar a instrução INPUT para atribuir valores à variáveis uma a uma, você pode usar a instrução DATA para armazenar uma lista de números ou strings na memória.

Para obter os dados da lista de instrução DATA, você deve usar a instrução READ. Exemplo:

```
10 READ A,B,C,D,E,F      (note que A,B,C,D,E e F compõe
                          uma lista de variáveis)
20 READ G                (aqui há somente uma variável
                          após o READ)
30 DATA 1,2,3,4,5,6,7
RUN
```

O computador atribui os dados a cada variável de acordo com a ordem das variáveis e a ordem dos dados. Exemplo:

```
A=1, B=2, C=3, D=4, E=5, F=6 e G=7
```

A instrução DATA pode ser colocada em qualquer lugar dentro do programa, apesar de normalmente ser colocada no final do mesmo. Você pode dividir uma longa lista de dados em várias instruções DATA, pois o computador irá tratá-las como uma instrução DATA simples, assumindo a ordem dos dados de acordo com a ordem em que as instruções DATA são apresentadas. Digite o seguinte:

```
10 DATA 1,2
20 DATA 3,4
30 DATA 5,6,7
40 READ A,B,C,D,E,F,G
50 PRINT A,B,C,D,E,F,G
RUN
```

Agora digite novamente a linha 50 da seguinte forma:

```
50 PRINT A;B;C;D;E;F;G
```

e execute o programa para comparar ao anterior.

O computador divide cada linha em campos de 14 caracteres de comprimento. Se os dados a serem exibidos forem separados por vírgulas, o computador exibe cada dado em um campo. Se eles são separados por ponto e vírgula, o computador exibe cada valor imediatamente após o anterior. Veja este exemplo com cadeia de dados e variáveis tipo cadeia.

```
NEW
10 READ X$,Y$,Z$
20 DATA " JOGOS ", " EDUCACAO ", " ORGANIZACAO "
30 PRINT "O COMPUTADOR PODE SER USADO PARA ";X$;Y$
    "E"; Z$
RUN
```

NOTA: o número de dados deve ser igual ou maior do que o total de variáveis das instruções READ, a não ser que você use uma instrução RESTORE para reutilizar os dados.

Quando uma instrução RESTORE é executada, a próxima instrução READ irá ler o primeiro dado da primeira instrução DATA do programa. Exemplo:

```
NEW
10 READ A,B
20 PRINT (A+B)/2
30 READ C
40 RESTORE
50 READ D
60 PRINT (A+B+C+D)/4
70 DATA 80, 120, 115
80 DATA 1, 2, 3
RUN
```

Você achará a instrução RESTORE útil quando começar a usar as técnicas dos loops de programa no capítulo 7. Tente usar RESTORE com um número de linha, tal como RESTORE 80.

PARA LISTAR UM PROGRAMA

Suponhamos que você digitou um programa e deseja revisá-lo antes de executá-lo para ver se não cometeu nenhum erro. Para isso, você deve digitar LIST e pressionar <RETURN> e o programa será apresentado na tela com o número das instruções em ordem crescente, mesmo que você não o tenha feito nessa ordem. Veremos mais sobre este comando no capítulo 6.

NUMERANDO AUTOMATICAMENTE AS LINHAS

Seu computador tem uma característica que lhe permite inserir um programa sem se preocupar em digitar os números das linhas.

Digite AUTO no modo direto. O número da linha começará automaticamente a partir de 10. Experimente isso em qualquer um dos programas mostrados anteriormente. Depois de inserir todas as linhas do programa, você pode sair do modo AUTO através de CTRL-C, isto é, mantenha pressionada a tecla <CTRL> e tecle <C>. Estes são outros formatos do comando AUTO:

AUTO X - inicia automaticamente a numeração das linhas desde a linha X;

AUTO X-Y - inicia automaticamente a numeração das linhas desde a linha X até a linha Y.

COMENTARIOS EM UM PROGRAMA

Você pode usar a instrução REM para colocar comentários no seu programa, para que se saiba de que assunto se trata. O computador não executará a mensagem após REM, ela somente aparecerá quando você listar o programa. Você pode colocar esta instrução em qualquer parte do programa. Exemplo:

```
NEW
10 REM CALCULAR O QUADRADO
20 INPUT "DIGITE 4 NUMEROS"; A,B,C,D
30 PRINT "OS QUADRADOS DOS NUMEROS SAO: "; A^2;
    B^2;C^2; D^2; "RESPECTIVAMENTE"
```

OS COMANDOS END E STOP

A instrução END é útil para isolar o programa principal de suas sub-rotinas (veja mais sobre sub-rotinas no capítulo 9). Se você tem certeza que não existe nenhuma linha de programa após a instrução END, você pode omiti-la. Exemplo:

```
10 REM EXPRESSAO MATEMATICA
20 X=3
30 PRINT X*9/2+8
40 END
```

Você pode interromper a execução de um programa em uma ou mais posições e examinar se existem erros antes do final do mesmo. A instrução STOP é usada para este propósito. Exemplo:

```
10 A=5
20 PRINT A-5*10/2+A
30 STOP
40 PRINT 5+A*2/3
```

Depois de checar, você pode apagar a instrução STOP do programa. Para apagar uma linha, simplesmente digite o número da mesma e pressione <RETURN>.

Depois de parar, o programa só pode continuar a execução se for utilizada a instrução CONT seguida de <RETURN>. Use este comando no último programa exemplo.

A INSTRUÇÃO CLEAR

Enquanto a instrução NEW apaga o que estiver na memória, a instrução CLEAR zera todas as variáveis numéricas e matrizes, se existir e, anula todas as variáveis string. Exemplo:

```
10 A=10 : B=20
20 PRINT A+B
30 CLEAR
40 PRINT A+B
50 C=30 : D=40
60 PRINT C+D
```

O resultado da linha 40 é zero, pois todas as variáveis definidas antes da instrução CLEAR são anuladas.

CAPÍTULO 6

EDITANDO SEU PROGRAMA

Este capítulo mostra a você como revisar e corrigir seu programa. A primeira metade deste capítulo é muito fácil de ser seguida, o restante é para aqueles que já tem alguma experiência em computação ou aqueles que pretendem aprimorar-se com mais técnicas.

INFORMAÇÕES ADICIONAIS SOBRE O COMANDO LIST

Anteriormente, mostramos a você como listar um programa completo. Você pode na realidade usar o comando LIST numa variedade de formatos tais como os exemplos abaixo:

```
LIST      mostra o programa inteiro
LIST X    mostra a linha X do programa
LIST X-Y  mostra o programa desde a linha X até a linha Y
LIST X-   mostra o programa desde a linha X até o fim do programa
LIST -Y   mostra o programa desde o começo até a linha Y
```

Tente os diferentes formatos com qualquer programa. Durante a listagem de um programa, você pode pressionar CTRL-S para fazer uma pausa a fim de examinar o programa, especialmente um programa longo. Para continuar a listagem depois da pausa, pressione qualquer tecla.

NOTA: CTRL-S é também usado para fazer uma pausa durante a execução de um programa.

APAGAR PARTE DA LINHA CORRENTE

Caso você queira apagar algo em uma linha que estiver digitando, você deve usar a tecla <RUBOUT>. Ela é usada para apagar um caracter ou espaço imediatamente antes do cursor. Digite esta tecla quantas vezes for necessário.

Para apagar a linha corrente inteira, basta você pressionar a tecla @. A linha continuará aparecendo na tela, mas já estará apagada na memória.

Caso você queira apagar uma linha de programa como já foi dito anteriormente, simplesmente digite o número da mesma e pressione <RETURN> e ela será anulada da memória.

INSERIR UMA NOVA LINHA

Se você quiser inserir uma ou mais linhas de programa entre as linhas X e Y, use um número de linha entre X e Y e digite a informação desejada. Exemplo:

```
10 PRINT "EDGARD"  
20 PRINT "DIRCEU"
```

Você poderá inserir qualquer instrução, desde que não estejam os números 10 e 20. Exemplo:

```
11 PRINT "EDUARDO"  
12 PRINT "FLAVIA"
```

Agora liste (LIST) o programa para ver o que aconteceu.

CORRIGIR UMA LINHA JÁ EXISTENTE

Se você tiver cometido algum erro em uma linha e já tiver teclado <RETURN>, para corrigi-la, você deve digitar novamente o número da linha e a instrução correta inteira ou utilizar o comando EDIT, afinal, é desagradável ter que digitar uma ou mais vezes uma linha já existente. Usando o comando EDIT junto com os subcomandos para fazer correções, você evita isso. No modo EDIT, todas as teclas de comandos pré-definidos (vide capítulo 2) não são executadas.

Para entrar no modo de edição, digite o seguinte:

```
EDIT X <RETURN> (onde o X é o número da linha na  
qual você deseja fazer correções).
```

Então pressione várias vezes a tecla de espaço para que o conteúdo da linha seja mostrado na tela, ou utilize a função L.

O conteúdo original daquela linha se mantém inalterado.
Agora digite este exemplo:

```
NEW
10 PRINT "COMO REVISAR UM PROGRAMA"
20 PRINT "QUAL LINHA VOCE QUER CORRIGIR?"
30 PRINT "TODAS AS LINHAS ERRADAS"
40 PRINT 30+40
50 PRINT 1+2+3+4+5
```

A FUNÇÃO L

A função L mostra o conteúdo da linha a ser revisada.

Suponhamos que você queira revisar a linha 10. Digite:

```
EDIT 10 <RETURN>
```

A tela mostrará:

```
10
```

agora digite L (não há necessidade de teclar <RETURN>). A tela mostrará:

```
10 PRINT "COMO REVISAR UM PROGRAMA"
10
```

Se você não tem nada a corrigir na linha 10, somente tecle <RETURN>. O conteúdo da linha 10 é mostrado na tela e o modo de Edição é abandonado. A linha 10 permanece inalterada.

A FUNÇÃO I

Esta função é usada para inserção de caracteres numa linha. Suponhamos que na linha 10 você tenha digitado PRGRAMA ao invés de PROGRAMA e naturalmente, quer corrigir o erro. Então digite:

```
EDIT 10 <RETURN>
```

a resposta será

```
10
```

Agora pressione a tecla <SPACE> várias vezes até que o cursor pare no lugar onde será inserida a letra, assim:

```
10 PRINT "COMO REVISAR UM PR
```

então tecle <I> e em seguida a letra O. Depois de digitada a letra O, pressione <RETURN> e a correção estará feita.

A FUNÇÃO D

Esta função é usada para anular caracteres. Por exemplo, se na linha 40 você tivesse digitado PRINMT ao invés de PRINT. Então, para corrigir digite:

```
EDIT 40 <RETURN>
```

e a tela mostrará

```
40
```

então digite <L> e depois pressione a barra de espaço aos poucos, até que a tela mostre:

```
40 PRIN
```

Tecla <D>, a tela mostrará

```
40 PRIN/M/
```

A letra M foi apagada, por isso, a mesma é mostrada entre barras. Caso você quisesse apagar mais alguma letra, você teria que digitar <D> novamente, ou <SPACE> para continuar a mostrar o conteúdo da linha. Para sair do modo EDIT, pressione <RETURN>.

A FUNÇÃO S

Esta função deve ser utilizada para substituir um caracter por outro. Digamos que na linha 30 você digitou TODOS ao invés de TODAS. Para corrigir digite:

```
EDIT 30 <RETURN>
```

```
30
```

agora digite <L> e vá pressionando aos poucos a tecla <SPACE> até que a tela apresente:

```
30 PRINT "TOD
```

então digite <A> e depois a letra R; e para sair do modo EDIT tecle <RETURN>.

A FUNÇÃO H

Esta função é usada para apagar todos os caracteres a partir do cursor até o fim da linha, e então inserir novos. Por exemplo, vamos mudar o conteúdo da linha 50 PRINT 1+2+3+4+5 para 2*4*6*8 da seguinte forma, digite:

```
EDIT 50 <RETURN>  
50
```

pressione a tecla <SPACE> até a tela ficar assim:

```
50 PRINT
```

agora tecle <H> e depois 2*4*6*8 então <RETURN>. Liste a linha 50 para ver se a troca foi feita corretamente.

A FUNÇÃO X

A função X mostra todos os caracteres a partir de onde o cursor estiver até o fim da linha, entrando automaticamente no modo de inserção. Por exemplo, digite:

```
EDIT 10  
10
```

agora tecle <SPACE> até a tela mostrar isso:

```
10 PR
```

então, pressione X e a tela mostrará a linha inteira:

```
10 PRINT "COMO REVISAR UM PROGRAMA"
```

A FUNÇÃO Q

Esta função permite sair do modo EDIT sem fazer qualquer alteração na linha chamada. Por exemplo, se você digitou

```
EDIT 30 <RETURN>
```

e depois de fazer algumas alterações resolveu manter a linha original, simplesmente pressione <Q> e você sairá do modo EDIT, e a linha continuará sem alteração.

CAPÍTULO 7

LOOPS, DESVIOS E DECISÕES

É muito frequente acontecer execuções repetitivas de instruções similares ou idênticas. Nestes casos, é conveniente a utilização de loops (laços) e desvios. A execução de um programa pode se desviar para outras partes do programa principal, dependendo de certas condições serem preenchidas ou não. O computador poderá comparar números, expressões ou strings para tomar essa decisão.

FOR...TO...STEP...NEXT

Tente digitar os exemplos a seguir e você descobrirá como é fácil de trabalhar com "loops".

Exemplo 1

```
-----  
10 FOR A=1 TO 100 STEP 2  
20 PRINT A;  
30 NEXT A  
RUN
```

agora redigite a linha 10 da forma descrita abaixo:

```
10 FOR A= 1 TO 100
```

e execute (RUN) o programa novamente.

Se o incremento de A for igual a +1, você pode omitir a instrução STEP, caso contrário, deve-se especificar o valor do passo após o STEP, que podem ser números negativos ou fracionários.

Agora, mude a linha 20 para

```
PRINT A
```

Execute o programa e veja o que acontece.

Exemplo 2

```
-----
10 FOR B=40 TO 0 STEP-4
20 PRINT B^2
30 NEXT
RUN
```

O nome da variável após o NEXT pode ser omitido, mas em programas que possuam muitos loops é melhor fazer uso do mesmo.

Exemplo 3

```
-----
10 HOME
20 INPUT "DIGITE HORA E MINUTO";H,M
30 GOSUB 140
40 FOR HI = H TO 23 -----
50 FOR MI = M + 1 TO 59 ----- !
60 FOR SI = 1 TO 24637 ----- !!
70 NEXT SI ----- !!
80 GOSUB 150 ----- !!
90 NEXT MI ----- !
100 M= -1 ----- !
110 NEXT HI -----
120 H = 0
130 GOTO 40
140 HI = H:MI=M
150 HOME:PRINT:PRINT:PRINT:PRINT TAB(12) HI;":":MI
160 RETURN
```

O loop SI está dentro do "laço MI" que por sua vez, está dentro do loop HI. Portanto NEXT SI deve vir antes de NEXT MI, que deve preceder NEXT HI, pois eles não podem se cruzar.

GOTO

A instrução GOTO faz com que o computador desvie a execução do programa para uma linha especificada. Exemplo:

```
10 Y=Y+1
20 PRINT Y;
30 GOTO 10
RUN
```

O exemplo acima mostra um método para formar um loop infinito com a instrução GOTO. Você também pode usar a instrução GOTO sem formar um loop, desviando a execução para outra parte do programa sem retornar. Para parar a execução do programa, pressione <CTRL> <C>.

IF . . . GOTO

Tente este exemplo:

```
10 INPUT "DE-ME DOIS NUMEROS";A,B
20 IF A=B GOTO 50
30 PRINT-"A E DIFERENTE DE B"
40 END
50 PRINT "A E IGUAL A B"
```

O programa acima faz um desvio somente se a condição após IF for satisfeita.

IF . . . THEN

Esta instrução é similar a IF..GOTO, exceto que a instrução após o THEN pode ser tanto um número de linha quanto uma outra instrução BASIC. Exemplo:

```
10 REM ADIVINHE NUMERO
20 PRINT "PENSE UM NUMERO DE 1 A 9"
30 PRINT
40 PRINT "EU ADIVINHAREI SEU NUMERO"
50 A=1
60 PRINT "O NUMERO E";A
70 PRINT
80 INPUT "ACERTEI (S/N)";K$
90 IF K$ = "S" THEN 130
100 IF K$ = "N" THEN 113
110 PRINT "VOCE DEVERA ESCREVER SOMENTE S OU N": GOTO
    80
113 A=A+1
115 IF A=10 THEN PRINT "EU NAO JOGO COM TRAPACEIROS":
    END
120 PRINT "EU VOU TENTAR OUTRA VEZ":GOTO 60
130 PRINT "EU SABIA QUE ACERTAVA":END
```

ON...GOTO

Esta instrução desvia a execução do programa para um dos números de linha especificados, dependendo do valor da expressão depois da instrução ON. Exemplo:

```
10 PRINT "GOTO COM MULTI DESVIOS"
20 INPUT "DIGITE UM NUMERO ENTRE 0 E 4 ";X
30 ON X GOTO 40,50,60,70
40 PRINT "SE X=0 OU > 4, A EXECUCAO CONTINUA NA LINHA
    SEGUINTE ":END
50 PRINT "SE X=1, A EXECUCAO DESVIA PARA O PRIMEIRO
    ITEM APOS O GOTO":END
60 PRINT "SE X=2, A EXECUCAO DESVIA PARA O SEGUNDO
    ITEM APOS O GOTO":END
70 PRINT "SE X=3, A EXECUCAO DESVIA PARA O TERCEIRO
    ITEM APOS O GOTO":END
80 PRINT "SE X=4, A EXECUCAO DESVIA PARA O QUARTO
    ITEM APOS O GOTO":END
```

Caso o valor da variável (neste caso X) seja 0 ou maior que o número de linhas relacionados, a execução do programa continuará na linha seguinte a da instrução ON GOTO.

OPERADORES RELACIONAIS

Para decisões do computador, se torna necessário comparar 2 números ou expressões. Números e/ou expressões podem ser comparadas usando operadores de relação e descobrindo se a relação é verdadeira ou falsa.

Os operadores de relação são:

=	igual
<> ou <>	diferente
<	menor que
=< ou <=	menor ou igual a
>	maior que
>= ou =>	maior ou igual a

Exemplo:

```
10 INPUT "DIGITE DOIS NUMEROS"; A,B
20 IF A => B THEN 50
30 PRINT A; "E MENOR QUE "; B
40 GOTO 10
50 IF A > B THEN PRINT A; "E MAIOR QUE ";B:GOTO 10
60 PRINT "OS DOIS NUMEROS SAO IGUAIS"
70 GOTO 10
```

Tente escrever programas similares usando outros operadores de relação.

OPERADORES LÓGICOS

São aqueles que executam operações lógicas, ou booleanas, em valores numéricos. Como nos operadores relacionais, eles são usados também em decisões, dentro de um programa. Operadores lógicos são normalmente usados para conectar duas ou mais relações, nos retornando um valor falso ou verdadeiro. Exemplo:

```
10 INPUT "DIGITE 3 NUMEROS"; A,B,C
20 IF A>B AND B>C THEN PRINT A; "E O MAIOR NUMERO":
   GOTO 50
30 IF A>B OR B>C THEN PRINT A; "OU";B; "E O MAIOR
   NUMERO":GOTO 50
40 IF A=B AND B=C THEN PRINT "OS TRES NUMEROS SAO
   IGUAIS":GOTO 50
50 PRINT "ESTE PROGRAMA SERVE PARA MOSTRAR O USO DOS
   OPERADORES LOGICOS"
60 LIST
```

Tente escrever programas simples que contenham relações tais como: $A < B$ ou $A < C$ ou $NOT A = 1$.

ORDEM COMPLETA DE PRECEDÊNCIA

A ordem completa de prioridade em operações numéricas é a seguinte:

- 1º) () - expressões dentro de parênteses
- 2º) . ^ - exponenciação
- 3º) - - negação
- 4º) * e / - multiplicação e divisão
- 5º) + e - - adição e subtração
- 6º) relações numéricas - todas com a mesma prioridade
 - = - igual
 - <> - diferente
 - > - maior que
 - < - menor que
 - >= - maior ou igual a
 - <= - menor ou igual a
- 7º) NOT
- 8º) AND
- 9º) OR

Observação: quando tivermos duas ou mais operações com o mesmo grau de prioridade, o computador as executará da esquerda para a direita.

CAPÍTULO 8

SUB-ROTINAS

O QUE É UMA SUB-ROTINA

Em um programa longo, normalmente é preciso efetuar uma mesma sequência de instruções várias vezes, tal como preparar valores para cálculos posteriores. Para fazer isso, o programa desvia do fluxo principal indo executar sequência de instruções. Após isto, a execução retorna ao programa principal. Esta pequena sequência é chamada de "sub-rotina". Você verá mais a este respeito durante este capítulo.

AS INSTRUÇÕES GOSUB E RETURN

Quando uma instrução GOSUB é executada, o programa desvia para a sub-rotina começada na linha especificada pelo GOSUB. No final da sub-rotina, você deve adicionar a instrução RETURN pois ela indica o final da sub-rotina. De outra forma, a execução não voltará ao programa principal.

NOTA: a instrução RETURN é diferente da tecla <RETURN>. Você tem que escrever a palavra RETURN ou usar o <SHIFT> <Z>.

Exemplo:

```
10 FOR I=1 TO 3
20 INPUT "DIGITE UM NUMERO";X
30 PRINT "IRA PARA A SUB-ROTINA"
40 GOSUB 90
50 PRINT "SE X = ";X;
60 PRINT "O VALOR DA FUNCAO Y E ";Y
70 NEXT
80 END
90 Y=X^3+2*X^2-30*X+1
100 PRINT "ESTA SUB-ROTINA FOI EXECUTADA ";I;" VEZES"
110 RETURN
```

Observação: a instrução END na linha 80 evita que ao terminar o programa, que o mesmo execute novamente a sub-rotina.

SUB-ROTINA CONDICIONAL MÚLTIPLA

A instrução ON..GOSUB desvia o programa principal para diferentes sub-rotinas. Seu formato é:

```
ON X GOSUB N1, N2, N3 etc
```

Se X=1, o programa vai para a sub-rotina começando da linha N1. Se X=n, o programa vai para a sub-rotina começando do n-ésimo número de linha depois da palavra GOSUB. Exemplo:

```
10 PRINT "SUB-ROTINA MULTIPLA"  
20 FOR A=1 TO 3  
30 ON A GOSUB 70,80,90  
40 PRINT "-SUB-ROTINA"  
50 NEXT  
60 END  
70 PRINT "ESTA E A PRIMEIRA";:RETURN  
80 PRINT "ESTA E A SEGUNDA";:RETURN  
90 PRINT "ESTA E A TERCEIRA";:RETURN
```

CAPÍTULO 9

FUNÇÕES NUMÉRICAS

Neste capítulo, serão mostradas todas as funções matemáticas que o seu MC-1000 pode executar.

FUNÇÕES TRIGONOMETRICAS

SIN(X) - fornece o valor do seno de X. O valor de X é em radianos.

Exemplo:

```
10 REM DESCUBRA O VALOR DE SENO DE 1.1
20 PRINT "O SENO DE 1.1 É "; SIN (1.1)
```

Analogamente,

COS(X) - fornece o valor do cosseno de X, onde X é em radianos

TAN(X) - fornece o valor da tangente de X em radianos

ATN(X) - fornece o valor arcotangente de X. O resultado será dado em radianos.

ALGUMAS FUNÇÕES MATEMATICAS

ABS(X) - Fornece o valor absoluto de X.

EXP(X) - Fornece o valor e elevado a potência X, onde e = 2.71828 e $X \leq 87.33365$.

INT(X) - Fornece o mais próximo número inteiro menor ou igual a X.

LOG(X) - Fornece o logaritmo de X na base 10 ($X > 0$).

SQR(X) - Fornece a raiz quadrada de X ($X \geq 0$).

SGN(X) - Fornece o sinal de X. Se $X < 0$, SGN(X) = -1. Se $X = 0$, SGN(X) = 0. Se $X > 0$, SGN(X) = 1.

A FUNÇÃO RND

A função RND gera um número aleatório entre 0 e 1, de acordo com as regras abaixo. Esta função é particularmente útil em programas de jogos.

RND(X) - Fornece um número entre 0 e 1.

Se $X > 0$ (qualquer número positivo), o resultado será um número aleatório.

Se $X < 0$ (qualquer número negativo), a função sempre retorna com o mesmo número aleatório.

Se $X = 0$, o resultado da função será o último número gerado por RND(X). Caso nenhum RND(x) tenha sido executado, a resposta será .811635.

Exemplo:

```
PRINT RND(3)
PRINT RND(8)
PRINT RND(0)
PRINT RND(-17)
PRINT RND(-9)
```

A INSTRUÇÃO DEF FN

É muito trabalhoso construir todas as funções matemáticas para o computador. Para isso, você pode usar a instrução DEF FN para definir suas fórmulas ou funções não incluídas em seu micro. Exemplo:

```
10 DEF FN Y(X) = X^2+3*X-7
15 X=2
20 PRINT FN Y(X)
```

Outro exemplo:

```
10 DEF FN Y(X) = ATN(X/SQR(-X*X+1))
20 X = 0.5
30 PRINT FN Y(X)
```

NOTA: aqui Y(X) nos fornece o valor do arco seno de X.

CAPÍTULO 10

SOM E MÚSICA

Seu computador pode gerar músicas e sons através de 3 canais, cada um com 8 oitavas. Você pode compor músicas ou simular vários sons, como aqueles nos video games.

AS INSTRUÇÕES TEMPO E SOUND

A instrução TEMPO é usada para definir o efeito envelope, compasso e o canal de som utilizado. O formato é:

TEMPO A,B,C

onde A define a frequência do envelope e tem valores de 0 a 255. Quanto menor o valor de A, mais curto é o ciclo envelope.

B tem um valor de 0 a 3 para definir o compasso do som.
B=0 tem o compasso mais rápido, e B=3 o mais lento.
C define um dos três canais de som: C=1, C=2 e C=3 correspondem aos canais 1, 2 e 3 respectivamente.

A instrução SOUND define o tom da nota, e a posição dos intervalos (duração) e o canal utilizado SOUND X,Y,Z.

X define o tom da nota.

Existem ao todo 96 notas em 8 oitavas. Na próxima seção, você verá uma tabela completa de códigos para notas.

Y define a duração da nota. Ela tem valores de 0 a 255.
Y=0 tem a nota mais curta e Y=255 a mais longa.
Z tem valores 1, 2 ou 3 correspondentes aos canais de som 1, 2 e 3 respectivamente.

TABELA DE CÓDIGOS PARA NOTAS MUSICAIS

TEMPO

! ESCALA !	! BAIXA !				! VELOCIDADE !			! ALTA !		
	! 4 !	! 3 !	! 2 !	! 1 !	! REGULAR !			! 1 !	! 2 !	! 3 !
! Do(1) !	! 16 !	! 32 !	! 48 !	! 64 !	! 80 !	! 96 !	! 112 !	! 128 !	!	!
! #Do(#1) !	! 17 !	! 33 !	! 49 !	! 65 !	! 81 !	! 97 !	! 113 !	! 129 !	!	!
! Re(2) !	! 18 !	! 34 !	! 50 !	! 66 !	! 82 !	! 98 !	! 114 !	! 130 !	!	!
! #Re(#2) !	! 19 !	! 35 !	! 51 !	! 67 !	! 83 !	! 99 !	! 115 !	! 131 !	!	!
! Mi(3) !	! 20 !	! 36 !	! 52 !	! 68 !	! 84 !	! 100 !	! 116 !	! 132 !	!	!
! Fa(4) !	! 21 !	! 37 !	! 53 !	! 69 !	! 85 !	! 101 !	! 117 !	! 133 !	!	!
! #Fa(#4) !	! 22 !	! 38 !	! 54 !	! 70 !	! 86 !	! 102 !	! 118 !	! 134 !	!	!
! So(5) !	! 23 !	! 39 !	! 55 !	! 71 !	! 87 !	! 103 !	! 119 !	! 135 !	!	!
! #So(#5) !	! 24 !	! 40 !	! 56 !	! 72 !	! 88 !	! 104 !	! 120 !	! 136 !	!	!
! La(6) !	! 25 !	! 41 !	! 57 !	! 73 !	! 89 !	! 105 !	! 121 !	! 137 !	!	!
! #Ls(#6) !	! 26 !	! 42 !	! 58 !	! 74 !	! 90 !	! 106 !	! 122 !	! 138 !	!	!
! Si(7) !	! 27 !	! 43 !	! 59 !	! 75 !	! 91 !	! 107 !	! 123 !	! 139 !	!	!

Exemplos:

```

10 TEMPO 100,2,1
15 HOME
20 FOR X=1 TO 7
30 READ A,B,C
40 SOUND A,B,C
50 FOR I=1 TO 250:NEXT I
60 NEXT X
70 RESTORE
80 GOTO 20
90 DATA 80,5,1,82,5,1,84,5,1,85,5,1
100 DATA 87,5,1,89,5,1,91,5,1

```

Para parar, pressione CTRL-C.

NOTA: use a instrução TEMPO somente uma vez por canal, a não ser que você queira redefinir o compasso do som.

CAPÍTULO 11

GRAFICOS E CORES

CORES

Seu computador pode mostrar 4 cores diferentes (incluindo o fundo) na tela da TV. Quando você seleciona: GR, nós visualizamos 3 cores. Existem 4 códigos para representar as 4 cores (obs.: GR aciona o modo gráfico):

0 = fundo
1 = amarelo
2 = azul
3 = vermelho

O formato geral para definir as cores é:

COLOR = código da cor

Veja os exemplos mais adiante.

TEXTOS E GRAFICOS

Quando você liga ou tecla <SHIFT> <RESET> no seu computador, a tela está no modo normal de texto. O computador pode também mostrar informações no modo gráfico de Alta e Baixa Resolução.

MODOS DE TEXTO

Você pode usar a instrução TEXT para selecionar o modo texto, usando o seguinte formato:

TEXT <RETURN>

Os caracteres somente podem ser vistos no modo texto. Para usar o modo gráfico, use a instrução GR ou HGR nos seguintes formatos:

GR <RETURN>

Este é o formato do modo gráfico de Baixa Resolução com 128*64 blocos e 4 cores. Quando você entrar no modo GR, lembre-se de ajustar a cor que desejar, caso contrário, o computador usará o código de cor que você ajustou anteriormente. Exemplo:

```
10 GR
20 COLOR=2
30 PLOT 0,0 TO 125,60
```

NOTA: a instrução PLOT será mostrado mais adiante.

Você também pode usar o modo de execução direta para entrar em modo gráfico apenas digitando GR, depois COLOR=2 e em seguida os comandos PLOT para desenhar.

Para usar o modo gráfico de Alta Resolução use o seguinte formato:

```
HGR <RETURN>
```

Este modo trabalha com 256*192 pontos e não há seleção de cores (preto e branco).

INSTRUÇÕES PLOT E UNPLOT

Estas instruções são usadas nos modos gráficos para desenhar na tela.

FORMATO 1

```
PLOT X,Y
```

Esta instrução colocará na tela um ponto locado com coordenadas X,Y.

Digite a linha 30 do programa anterior da seguinte maneira:

```
30 PLDT 64,32
```

e surgirá um ponto no centro da tela.

FORMATO 2

PLOT X1,Y1 TO X2,Y2 TO...TO Xn, Yn

Esta instrução colocará uma linha reta do ponto (X1,Y1) para o ponto (X2,Y2), depois uma linha de (X2,Y2) para (X3,Y3), e assim por diante até o ponto (Xn,Yn).

Naturalmente n poderá ser maior ou igual a 2, mas a instrução não pode exceder uma linha de programa de 255 caracteres.

Agora digite a linha 30 assim:

```
30 PLOT 0,5 TO 0,60 TO 125,60 TO 125,5
```

e na tela surgirão 3 barras, uma em cada lado e uma outra na parte inferior.

FORMATO 3

PLOT TO X2,Y2

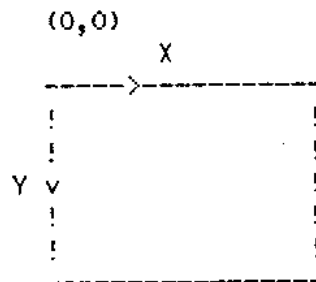
Esta instrução colocará na tela uma linha reta iniciando no último ponto traçado até o ponto X2,Y2. Se nenhum ponto foi assinalado previamente, então a linha partirá do ponto de origem (0,0) e irá até o ponto (X2,Y2).

Inclua a linha 40 no programa:

```
40 PLOT TO 0,5
```

e então você terá um quadrado completo.

NOTA: o sistema de coordenadas acima tem a origem (0,0) localizada no canto superior esquerdo. A coordenada Y conta os pontos para baixo e X para a direita.



No modo semigráfico $0 \leq X \leq 127$; $0 \leq Y \leq 63$
No modo gráfico $0 \leq X \leq 255$; $0 \leq Y \leq 191$

Exemplos:

```
10 HGR
20 FOR X=0 TO 255
30 PLOT 0,0 TO X,191
40 NEXT
```

```
NEW
10 HGR
20 FOR T=1 TO 16 STEP .10
30 X=SIN(T*2+1.57)*120+125
40 Y=SIN(T*2)*80+85
50 X1=SIN(T*3+1.57)*120+125
60 Y1=SIN(T*3)*80+85
70 PLOT X,Y TO X1,Y1
80 NEXT
```

A instrução UNPLOT tem os mesmos formatos de PLOT, porém UNPLOT tem a função de apagar um ponto ou uma linha.

AS INSTRUÇÕES DRAW E UNDRAW

DRAW X,Y nos fornece uma linha reta desde o ponto anterior, digamos: X0,Y0 para o ponto X1,Y1, onde $X1=X0+X$ e $Y1=Y0+Y$. Desta vez X,Y podem ter valores negativos.

Observe que a diferença entre PLOT e DRAW (ou UNPLOT e UNDRAW) é que a instrução PLOT usa coordenadas absolutas com origem em (0,0) e DRAW usa coordenadas relativas com origem no último ponto assinalado.

A instrução UNDRAW X,Y apaga uma linha usando as coordenadas relativas.

TROCA DE ORIGEM

Quando o computador entra no modo gráfico, a origem é pré-definida no ponto 0,0. Você deve usar a instrução SET X,Y se quiser redefinir uma nova origem no ponto X,Y.

DISPLAY NORMAL E INVERSE

Quando o micro é ligado, os caracteres aparecem no estado normal, isto é, caracteres claros sobre fundo escuro. Você pode também ter um display com as cores invertidas. Os comandos NORMAL e INVERSE são usados para inverter as cores na tela. Por exemplo, digite:

INVERSE <RETURN>

e digite alguns caracteres. Como você pode ver, os caracteres digitados são escuros sobre fundo claro. Agora digite:

NORMAL <RETURN>

e você está novamente com caracteres claros sobre fundo escuro. Este comando também pode ser usado no modo indireto.

A INSTRUÇÃO COLUMN

Seu computador mostra na tela 32 colunas x 24 linhas. Se você colocar o cartão de 80 colunas e digitar o comando "COLUMN 80", a tela passará a mostrar 80 colunas x 24 linhas.

Você só precisa digitar o comando "COLUMN 32" para retornar ao estado normal.

NOTA: se você esquecer de colocar o cartão de 80 colunas e digitar o comando "COLUMN 80", seu computador mostrará na tela a mensagem "?TI ERRO".

Caso você tente usar este comando com parâmetros diferentes de 32 ou 80, você terá como resposta a mensagem "?PI ERRO".

Confira qual foi seu erro e volte a trabalhar normalmente. Você poderá ter maiores informações a este respeito no manual que acompanha a placa de 80 colunas.

CAPÍTULO 12

STRINGS E MATRIZES

Nos capítulos 4 e 5, nós lhe mostramos o que são strings e como introduzi-las ou imprimi-las. Neste capítulo, introduziremos algumas funções e a idéia de matriz.

A FUNÇÃO ASC

Cada caracter do teclado, incluindo a tecla de espaço, tem um código chamado ASCII (American Standard Code for Information Interchange), que é um código numérico (variando de 0 a 255) para representar todos os caracteres, comandos, símbolos e tudo o que o computador "entende". A função ASC(X\$) mostra o valor ASCII do primeiro caracter da cadeia de caracteres X\$. Digite o seguinte:

```
10 FOR X=1 TO 6
20 READ A$
30 PRINT ASC(A$); "-ESTE E O CODIGO ASCII DA LETRA:";
  A$
40 NEXT
50 DATA "E", "D", "G", "A", "R", "D"
```

A FUNÇÃO CHR\$

Esta função é o oposto da função ASC. Você pode usar esta função para obter um caracter do teclado se você souber seu código ASCII:

```
10 FOR I = 0 TO 255
20 PRINT "CHR$ (";I;") = "; CHR$(I)
30 PRINT
40 NEXT
```

Outro exemplo:

```
10 FOR I = 1 TO 6
20 READ A
30 PRINT CHR$(A)
40 NEXT
50 DATA 76,73,76,73,65,78
```


CONCATENAÇÃO DE STRINGS

A concatenação de strings refere-se à "inclusão" ou encadeamento de duas ou mais strings de caracteres para formar uma mais longa. Exemplo:

```
10 A$ = "ED"  
20 B$ = "UAR"  
30 C$ = "DO"  
40 PRINT A$+B$+C$
```

NOTA: você pode usar o sinal "+" para somar strings, porém os sinais -, * e / não podem ser usados.

Veja outro exemplo:

```
20 FOR I = 1 TO 255  
30 A$ = A$ + "A"  
40 PRINT I : PRINT A$  
50 NEXT
```

Desta maneira, você pode formar uma nova cadeia com um máximo de 255 caracteres, incluindo espaços.

A FUNÇÃO LEN

A função LEN(X\$) mostra o comprimento da string dentro das aspas ou correspondente a variável. Os caracteres em branco também são contados. Exemplo:

```
PRINT LEN("CARLOS VALERIO")
```

a resposta será 14.

```
A$ = "12345AECDE"  
PRINT LEN(A$)
```

AS FUNÇÕES LEFT\$, RIGHT\$ E MID\$

As funções LEFT\$ e RIGHT\$ exibem os caracteres do extremo esquerdo e extremo direito, respectivamente e uma string. O número de caracteres desejados pode então ser especificado.

O formato geral é

LEFT\$(A\$,N)

onde A\$ é a string original e N é o número de caracteres desejados do extremo esquerdo de A\$. Se $N \geq \text{LEN}(A\$)$ então a string toda será a resposta. Exemplo:

```
10 INPUT "QUANTAS LETRAS TEM O SEU PRIMEIRO NOME";B
20 INPUT "DIGITE SEU NOME COMPLETO";A$
30 B$=LEFT$(A$,B): PRINT "POSSO TE CHAMAR DE ";B$:
   PRINT
40 N=LEN(A$)-B: C$=RIGHT$(A$,N):PRINT "OU DEVO
   CHAMA-LO DE DR. ";C$
```

Como você pode ver, RIGHT\$(A\$,M) mostra a string contando M caracteres da direita de A\$.

Você pode também usar caracteres que não estejam nos extremos da string usando a função MID\$. O formato geral é:

MID\$(A\$,X,Y)

onde X é a posição da qual você começa a contar os caracteres e Y é o número de caracteres a serem usados. Exemplo:

```
10 A$ = "BERNARDO"
20 PRINT MID$(A$,2,3): PRINT MID$(A$,4,5)
30 PRINT MID$(A$,1,5): PRINT MID$(A$,6,3)
```

NOTA: se você tentar obter mais caracteres do que os existentes na string, a função MID\$ responderá toda a string a partir do primeiro caracter indicado.

A FUNÇÃO STR\$

Ela converte um número em uma string. Por exemplo, o computador entende o número 101 como um valor numérico, mas se utilizarmos STR\$(101) o computador entenderá como uma string = "101".

Se o valor dentro dos parênteses é uma expressão numérica, a função STR\$ dará a representação do resultado final.

Por exemplo, a resposta da função STR\$(1+2+3) é a string "6" e não "1+2+3".

Exemplo:

```
10 INPUT "DIGITE DOIS NUMEROS";A,B
20 PRINT "A SOMA DOS NUMEROS E= ";A+B
30 PRINT "OU" STR$(A)+STR$(B)
40 PRINT "VOCE PODE VER A DIFERENCA!"
```

A FUNÇÃO VAL(X\$)

Ela fornece o valor numérico da string X\$, se o primeiro caracter de X\$ não for +, - ou um dígito, VAL(X\$)=0. Exemplo:

```
10 X$ = "23B7":Y$="-56ABC":Z$="ABC"
20 PRINT VAL(X$), VAL(Y$), VAL(Z$)
```

Nestes exemplos, as respostas serão:

```
23 -56 0
```

MATRIZES NUMERICAS E ALFANUMERICAS

Algumas vezes é útil dar um nome a um conjunto inteiro de números ou strings e no programa se referir ao conjunto ou a cada um de seus elementos. Este conjunto é chamado de "matriz". Ao se dimensionar a matriz, o computador aloca um bloco de memória para armazená-la, isto é, separa um espaço na memória especialmente para estes dados.

Para dar nome a uma matriz com elementos numéricos, siga as mesmas regras das variáveis numéricas. Para uma matriz com elementos alfanuméricos, siga as mesmas regras das strings (não esqueça o sinal \$ depois de cada nome).

A matriz mais simples é chamada matriz linear, na qual os elementos são listados em uma sequência ordenada. Exemplo: a matriz que segue é uma matriz linear a qual chamaremos A(X).

1.5,3.7,8.9,0.4,7.6,2.8,9

Para se referir a cada elemento de uma matriz, é necessário o uso de um índice após seu nome. Neste computador, a prática comum é começar o índice de 0. Portanto, no exemplo acima: A(0)=1.5, A(1)=3.7; A(2)=8.9; A(3)=0.4; A(4)=7.6, A(5)=2.8; A(6)=9.

A diferença entre os nomes das variáveis e dos nomes das matrizes é que os nomes das matrizes tem parenteses. Por exemplo AB e X\$ são nomes variáveis enquanto AB(Y) e X\$(3) são nomes de elementos de uma matriz.

```
10 A(0)=1.5: A(1)=3.7: A(2)=8.9
20 A(3)=0.4: A(4)=7.6: A(5)=2.8: A(6)=9
30 A$(0)="DAGOBERTO":A$(1)="ADAMIL":A$(2)="EDSON" :
   A$(3)= "SUZANE"
40 A$(4)="FREITAS":A$(5)="ADELIA":A$(6)="PRATTI"
50 FOR X=0 TO 6
60 PRINT A(X),A$(X)
70 NEXT
```

NOTA: você pode usar uma variável dentro dos parênteses da matriz para ter acesso ou chamar a matriz como no exemplo acima.

Uma outra maneira de preencher uma matriz é esta:

```
10 FOR X = 0 TO 6
20 READ A(X)
30 NEXT X
40 DATA 1.5,3.7,8.9,0.4,7.6,2.8,9
50 FOR K = 0 TO 6
60 PRINT A(K)
70 NEXT K
```

As linhas 10 a 40 preenchem a matriz. Você pode combinar o READ e o PRINT como se segue:

```
10 FOR X = 0 TO 6
20 READ A(X)
30 PRINT A(X)
40 NEXT
50 DATA 1.5,3.7,8.9,0.4,7.6,2.8,9
```

MATRIZES DE ORDEM SUPERIOR

Apesar de termos considerado apenas matrizes lineares, é muito frequente e útil expressar informações em matrizes bidimensionais ou matrizes tridimensionais (ou até de ordem superior de dimensão). Um exemplo de uma matriz bidimensional é a tabela musical no capítulo 10 deste manual.

O formato geral para uma matriz de ordem N é:

$$A(X_1, X_2, X_3 \dots X_n)$$

onde $X_1, X_2, X_3 \dots X_n$ são índices dos elementos da matriz A. Existem N índices para um elemento de uma matriz de n-ésima ordem. Exemplo:

```
10 FOR X=0 TO 2 : FOR Y = 0 TO 2
20 A(X,Y) = X+Y
30 PRINT X;Y;A(X,Y)
40 NEXT Y,X
```

O formato geral para a instrução DIM para uma matriz A de n-ésima ordem é:

$$\text{DIM A (d}_1, \text{d}_2, \text{d}_3 \dots \text{d}_n)$$

onde d_1 é o valor máximo para o 1º índice; d_2 é o valor máximo para o 2º índice e assim por diante.

Por exemplo, uma matriz tridimensional (3ª ordem) $A(X_1, X_2, X_3)$ pode ter dimensões:

$$\text{DIM A(2,2,3)}$$

na qual $d_1=2$, $d_2=2$, $d_3=3$ e a lista completa de elementos é ilustrada com o seguinte exemplo:

```
10 DIM A(2,2,3)
20 FOR X = 0 TO 2
30 FOR Y = 0 TO 2
40 FOR Z = 0 TO 3
50 A(X,Y,Z) = X+Y+Z
60 PRINT "A (";X;Y;Z;")=";A(X,Y,Z)
70 NEXT Z,Y,X
```

Observação: você pode somar duas ou mais matrizes que tenham a mesma ordem e mesma dimensão em cada índice.

CAPÍTULO 13

ARMAZENANDO E CARREGANDO PROGRAMAS EM FITA CASSETE

O GRAVADOR CASSETE

Você pode usar um gravador cassete comum para guardar e carregar seus programas em BASIC.

Ele deverá ser de boa qualidade com tomadas EAR e MIC (alguns fabricantes as chamam de EARPHONE ou MONITOR).

Para usá-lo, deixe-o com o controle de volume em 3/4 do máximo. Desligue ou abaixe os controles de TREBLE e BASS se houver. Se existir somente um controle de tom (TONE), deixe-o na posição média.

Para acoplar o seu gravador, siga as instruções abaixo.

- 1- Desligue seu micro. Nunca se deve conectar ou desconectar nenhum periférico a um computador ligado, pois isto pode danificar seu equipamento.
- 2- Junto com o seu MC-1000, você recebeu um cabo que tem dois plugs em cada ponta. Ligue o plug de uma das pontas no micro (um no conector EAR e a outra no MIC, ambos na lateral esquerda da console) e a outra no gravador. Certifique-se que o mesmo cabo está ligado no conector EAR nos dois aparelhos.

Caso seu gravador não armazene ou carregue programas, verifique se:

- a bateria está fraca;
- a fita é muito velha;
- o cabeçote do gravador está sujo;
- as posições dos botões de controle estão erradas;
- os terminais EAR e MIC estão conectados incorretamente;
- existe algum terminal solto.

OS COMANDOS SAVE E LOAD

Você deve usar o comando SAVE para armazenar seu programa em BASIC na fita e o comando LOAD para carregá-los de volta para o computador. Você deve dar um nome ao arquivo após os comandos SAVE e LOAD para identificar um programa BASIC. O nome do arquivo deve ser uma sequência alfanumérica de no máximo 5 caracteres, tal como 12345, ABC ou XY29. Exemplo:

```
10 REM PARA GUARDAR UM PROGRAMA
20 FOR A=1 TO 10
30 PRINT RND(A)
40 NEXT
SAVE 123
```

Antes de pressionar <RETURN> após o comando SAVE 123, você deve apertar as teclas RECORD e PLAY do gravador. O computador dará o sinal de OK quando o armazenamento estiver terminado.

Pare o gravador. Agora volte a fita para o início (ou pouco antes do início). Tecele <SHIFT> <RESET> e digite:

```
LOAD 123
```

antes de teclar <RETURN>, pressione a tecla PLAY do gravador.

Depois que o carregamento estiver terminado digite:

```
LIST <RETURN>
```

e veja o que acontece.

PARA GUARDAR E CARREGAR DADOS

O comando SAVE* é usado para armazenar somente os dados de um programa. Antes que o comando SAVE* seja executado, certifique-se que o programa tenha sido executado pelo menos uma vez. Além disto, os dados devem ser "dimensionados" como matrizes. Veja o exemplo abaixo:

```
10 DIM AB(30)
20 FOR I = 0 TO 30
30 AB(I) = I*I
40 PRINT AB(I)
50 NEXT
```

Depois de executar o programa uma vez, você deve escrever SAVE* AB e operar o gravador como descrito anteriormente.

NOTA: o nome da matriz a ser armazenada deve ser, separado do comando SAVE* por um espaço.

Veja outro exemplo:

```
10 DIM PQ(2,2)
20 FOR X = 0 TO 2
30 FOR Y = 0 TO 2
40 READ A
50 PQ(X,Y) = SQR(A): PRINT PQ(X,Y)
60 NEXT Y,X
70 DATA 1,2,3,4,5,6,7,8,9
RUN
```

Para armazenar os dados de uma matriz PQ, você deve digitar:

```
SAVE* PQ
```

e acionar o gravador antes de pressionar <RETURN> como descrito anteriormente.

Para carregar de volta o dado na forma de matriz, use o comando LOAD* (e acione o gravador da mesma forma que o comando LOAD).

Para o 1º exemplo, digite:

```
10 DIM AB(30)
20 LOAD* AB
RUN
```

NOTA: o nome da matriz deve também estar separado do comando LOAD* por um espaço.

Se quiser ver os dados após terem sido carregados, digite:

```
5 DIM AB(30)
10 LOAD* AB
20 FOR X=0 TO 30
30 PRINT AB(X)
40 NEXT
RUN
```

Depois você deverá pressionar a tecla PLAY do gravador e <RETURN>.

Para o 2º exemplo dado acima, digite:

```
10 DIM PQ(2,2)
20 LOAD* PQ
RUN
```

NOTA: matrizes alfanuméricas (strings) são rejeitadas pelos comandos SAVE* e LOAD*.

CAPÍTULO 14

INSTRUÇÕES RELACIONADAS AO SISTEMA

Neste capítulo, introduziremos mais instruções relativas ao sistema do MC-1000. É recomendável a leitura de livros técnicos para um aprimoramento nesta área.

A INSTRUÇÃO FRE

Esta instrução fornece o número de bytes da memória do computador livres para seu uso. Digite:

```
PRINT FRE(7)
PRINT FRE(6.5)
```

O número dentro do parênteses pode ser qualquer número. É um pseudo parâmetro.

MAPA DE MEMÓRIA

Cada byte da memória do computador possui um endereço, para que o conteúdo de cada byte possa ser lido ou gravado pelo programa ou pela unidade de controle no computador. Existem 2 tipos de memória: RAM (Random Access Memory) e ROM (Read Only Memory). O primeiro tipo é flexível tal que você pode gravar (introduzir) dados na memória ou ler dados dela. O dado estará perdido depois que você desligar o computador. O segundo tipo (ROM) armazena permanentemente todos os dados que foram gravados nela. Você não pode mudar o conteúdo na ROM. Existem 65536 endereços que você pode atribuir aos 2 tipos de memória. Nós chamamos esta atribuição de "mapa de memória".

A seguir temos um resumo do mapa de memória deste computador.

TIPO	ENDEREÇO	USO	
		BASIC	FITAS CASSETE DE JOGOS
RAM	0-FF	buffer de execução do programa DEBUG	
	100-1FF	buffer de execução do Monitor	
	200-2FF	buffer de linha	
	300-3D4	buffer de execução do interpretador BASIC	Área para programas Assembly
	3D5-BCFF	programa fonte BASIC	
	BD00-BFFF	stack e matriz do interpretador BASIC	
	ROM	C000-FFFF	Monitor e Interpretador BASIC

NOTA: a memória de vídeo é visível somente quando você a acessa. Para instruções detalhadas, informe-se no "Manual de Referência Técnica".

AS INSTRUÇÕES PEEK E POKE

Você pode ver o conteúdo de um endereço de memória usando a instrução PEEK. Exemplo:

```
10 A = PEEK (10000)
20 PRINT A
30 PRINT PEEK (11000)
```

Note que os endereços 11000 e 10000 são expressados no sistema de números decimais.

Você pode colocar dados em uma posição de memória usando a instrução POKE que tem o formato:

```
POKE A,D
```

onde A é o endereço em decimal e D o número decimal representando o dado. Veja este exemplo:

```
10 FOR X = 1 TO 10
20 POKE 16383+X, 64+X
30 D = PEEK (16383+X)
40 PRINT CHR$(D)
50 NEXT
```

O CURSOR

Quando o computador está no modo texto editando programas BASIC, você pode escolher um cursor piscante ou estável, através dos seguintes comandos:

```
cursor estável
-----
POKE 304,201 <RETURN>
```

```
cursor piscante
-----
POKE 304,195 <RETURN>
```

PORTAS DE ENTRADA E SAÍDA (E/S)

Em termos técnicos simples, um computador, consiste de uma unidade de processamento central (CPU), memória e alguns circuitos de interface que ligam a CPU e a memória a dispositivos externos que colocam ou retiram dados da memória. O aparelho de TV, o teclado, a impressora e o drive são todos dispositivos externos. O meio pelo qual os dispositivos externos são conectados ao computador é chamado de porta de E/S (ENTRADA/SAÍDA). A cada uma dessas portas, é atribuído um número de identificação. Neste computador, os números de identificação vão de 0 a 255, isto é, haverá um máximo de 256 portas de E/S.

A INSTRUÇÃO OUT

Com esta instrução, você pode enviar dados diretamente a uma Porta de E/S. O formato geral é:

OUT A,D

onde A é o número de identificação (número decimal interno) da porta $0 \leq A \leq 255$ e D é o valor representando o dado $0 \leq D \leq 255$.

A INSTRUÇÃO INP

Com esta instrução, você pode ler diretamente os dados numa porta de E/S. O formato geral é:

INP(A)

onde A é o número de identificação da Porta de E/S $0 \leq A \leq 255$.

A INSTRUÇÃO WAIT

É com certa frequência que dispositivos externos tal como uma impressora está ocupada fazendo algo quando o computador tenta enviar dados para ela. A instrução WAIT funciona como um semáforo para controlar o fluxo de sinais através das portas de E/S.

Existem dois formatos:

WAIT A,B

e

WAIT A,B,C

"A" é o número de identificação da Porta de E/S, B e C são números inteiros de 0 a 255. Você pode atribuir valores que sirvam para B e C para impor condições na instrução WAIT. A, B e C devem ser números decimais.

Para explicar como a instrução WAIT é executada:

d= o número binário representando o dado presente na porta A;
b= equivale ao número binário de B;
c= equivale ao número binário de C;

No primeiro formato: o computador executará a operação lógica $d \text{ AND } b$. Se o número binário resultante tem todos os 8 dígitos iguais a zero, o computador irá repetir a instrução WAIT. Se o número resultante tem pelo menos um dígito igual a "1", o computador irá executar a instrução que segue a instrução WAIT.

No segundo formato: o computador primeiro executa a operação lógica "ou exclusivo": $d \text{ XOR } c$.

LET e = d XOR c

Depois o computador continuará a operação lógica: $e \text{ AND } b$. Se o número binário resultante é zero, o computador repetirá a instrução WAIT. Se o resultado tem pelo menos um dígito igual a "1", o computador executará a próxima instrução.

CAPÍTULO 15

OUTROS COMANDOS E INSTRUÇÕES

A INSTRUÇÃO PR#

Se você quiser imprimir ou listar programas na impressora, você pode digitar:

```
PR#1: LIST <RETURN>
```

e o programa será listado na impressora. Depois disto, o computador reassume a condição PR#0 automaticamente que significa exibição na tela.

PR#1 é para LIST ou PRINT na impressora;
PR#0 é para LIST ou PRINT na tela;
PR#2 é para LIST ou PRINT em ambos (tela e impressora).

PARA DEPURAR UM PROGRAMA

Você pode depurar a execução de um programa usando a instrução TRON. Para voltar ao normal, use a instrução TROF. Exemplo:

```
10 TRON
20 FOR A = 1 TO 50
30 PRINT A;
40 NEXT
50 FOR B = 1 TO 30
60 PRINT A+B
70 NEXT
```

TABULAÇÃO

Como numa máquina de escrever; você pode fazer tabulações para exibir dados em posições específicas.

Neste computador, números positivos são precedidos de um espaço em branco, apesar de você não ver o sinal (+). Um espaço em branco é dado após cada número, então se um número positivo for impresso logo após outro número ou uma cadeia, existem pelo menos um espaço em branco entre eles. Nenhum espaço é automaticamente dado antes ou depois de uma cadeia impressa.

A INSTRUÇÃO TAB(N)

Você pode usar a instrução TAB para imprimir um número ou uma cadeia na posição do n-ésimo caracter horizontal desde a posição zero da extrema esquerda. $0 \leq N \leq 31$. Exemplo:

```
10 PRINT TAB(1);-1
20 PRINT TAB(1);1
30 PRINT TAB(1);"A"
40 PRINT TAB(5);6
```

A INSTRUÇÃO SPC(N)

Esta instrução imprime N espaços em branco desde a posição do extremo esquerdo $0 \leq N \leq 255$. Digite:

```
PRINT SPC(125);3
```

Para maiores informações sobre a distribuição na tela, veja o "Manual Técnico de Referência".

A INSTRUÇÃO WIDTH

Esta instrução é usada para definir o número de caracteres numa linha da tela (não de programa) ou numa impressora.

O formato geral é:

```
WIDTH n
```

onde $0 \leq n \leq 255$.

Se a largura da linha não é redefinida por esta instrução, a linha normal da tela tem ficará com 32 caracteres.

Se a tela da sua TV não mostra os caracteres do extremo direito, tente redefinir o comprimento da linha exibida através de:

WIDTH 29 <RETURN>

CAPÍTULO 16

DESENVOLVIMENTO DE SOFTWARE E DEPURACÃO (DEBUGING) EM LINGUAGEM DE MÁQUINA

Se você quer escrever software baseado nos códigos de máquina Z-80, você não precisa gastar milhões de cruzeiros para comprar um sistema de desenvolvimento de software. Este computador por si próprio funciona como uma ferramenta de desenvolvimento.

O COMANDO DEBUG

Quando você digita:

DEBUG <RETURN>

você para de usar o BASIC e passa a utilizar o programa do sistema monitor. Podendo então usar vários comandos para escrever programas em código de máquina. Os comandos são listados como segue:

COMANDO	FUNÇÃO
D	Display de dados
F	Colocar dados
S	Substituir dados na memória
M	Mudar o endereço do dado
G	Continuar a execução de programas (o mesmo que RUN em BASIC)
X	mostrar ou substituir dados em um ou todos os registradores
R	Similar ao subcomando X
Q	Retornar ao BASIC

O SUBCOMANDO D

Formato: DXXXX YYYY <RETURN>

A tela mostrará o conteúdo das localizações de memória de XXXX a YYYY.

NOTA: no modo DEBUG, os dados e endereços são sempre expressados em números hexadecimais. Você deve deixar um espaço vazio ou uma vírgula entre o endereço inicial e o final.

Se você cometer algum erro de digitação, você terá que redigitar o seu comando, pois o computador anula automaticamente a linha digitada. Quando os dados estão sendo mostrados, você pode fazer uma pausa pressionando CTRL-S e continuar o display pressionando qualquer tecla. Ou também pode parar a listagem através de CTRL-C. Para continuar, utilize o comando D a partir do ponto de parada. Quando o computador terminar de listar os dados, ele estará pronto para receber novos comandos.

NOTA: não deixe espaço após o "D"

Exemplo:

```
D100,500
D157F 2000
```

O SUBCOMANDO F

Formato: FXXXX YYYY DD <RETURN>

Você preenche todas as posições de XXXX a YYYY com o valor hexadecimal DD.

NOTA: não deixe nenhum espaço depois do "F".

Exemplo:

```
F0010,0020 2C3 <RETURN>
D0010 0020 <RETURN>
F0021 0021 3A <RETURN>
D0021,0024 <RETURN>
```

O SUBCOMANDO S

Formato: SXXXX <RETURN>

Você pode modificar o conteúdo do endereço de memória XXXX substituindo os dados já existentes por novos. Após a substituição do dado, o cursor salta automaticamente para o próximo endereço. Se você não quer fazer mais substituições, pressione a barra de espaço para sair do comando S. Exemplo:

S0005 <RETURN>

e substitua 0005 por C3, 0006 por AD e 0007 por FF para o dado original.

Agora tente:

D0005 0007

NOTA: caso queira manter o dado original, teclie <RETURN> e será como se você tivesse redigitado o mesmo.

O SUBCOMANDO M

Formato: MXXXX YYYY ZZZZ <RETURN>

Você repete o bloco de dados que começa na posição XXXX e vai até YYYY inclusive, para a nova posição começando em ZZZZ. O dado original permanecem intactos. Exemplo:

D0005 0018
M0008 000F 0018
D0008 001F

O SUBCOMANDO G

Este comando é equivalente ao comando RUN em BASIC.

Formato 1: GXXXX <RETURN>

O computador começa a execução a partir do endereço XXXX. Caso o mesmo não tenha sido especificado, a execução iniciará a partir do último endereço acessado.

Formato 2: GXXXX -YYYY <RETURN>

Após você ter teclado o endereço inicial XXXX e pressionado a barra de espaço uma vez, o computador mostra automaticamente um sinal "-", e você pode então teclar o endereço final YYYY.

Formato 3: GXXXX -YYYY -ZZZZ

Este é similar ao formato 2. É usado se o programa condicionalmente se bifurca em 2 endereços finais. A execução será interrompida em YYYY ou ZZZZ, dependendo de qual dos dois endereços a execução chegar primeiro.

O SUBCOMANDO X

Este comando é usado para mostrar ou trocar o conteúdo dos registradores na UCP (Unidade Central de Processamento).

Formato 1: X <RETURN>

O conteúdo de todos os registradores (A, B, C, D, E, F, H, I, L, M, X, Y, P e S) será mostrado. Tente digitar:

X <RETURN>

Formato 2: `Xr <RETURN>

onde r é um dos nomes de registradores acima. Por exemplo, digite:

XA <RETURN>

a tela mostrará A= AA- (assumindo que o conteúdo de A é AA).

Se você digitar BB antes de teclar <RETURN>, o conteúdo de AA será substituído por BB.

O SUBCOMANDO R

Este comando é o mesmo que o comando X, exceto que lida com os registradores A', B', C', D', E', F', I', L', M, X, Y, S e F.

FINALIZAR UMA SUB-ROTINA EM LINGUAGEM DE MÁQUINA

Sempre finalize uma subrotina em código de máquina usando o código de operação 0C9H e no final use o código de interrupção 0F7H (RST 6).

RETORNAR AO BASIC

Digite:

Q <RETURN>

O programa BASIC original, antes do uso da instrução DEBUG ainda existe e permanece intacto na memória.

PARA CHAMAR UMA SUB-ROTINA EM LINGUAGEM DE MÁQUINA

Em BASIC, você pode desviar para uma sub-rotina em código de máquina através da instrução CALL. Seu formato é:

CALL adr

onde adr é o endereço inicial da sub-rotina expressa no sistema de número decimal. Exemplo:

```
10 CALL 49169  
20 GOTO 10
```

CAPÍTULO 17

PROGRAMAS DA FITA CASSETE

Neste capítulo, você encontrará instruções de utilização dos cinco programas existentes na fita cassete que acompanha o seu MC-1000, sendo 4 jogos e um educacional.

Desenvolvidos em linguagem Assembly, proporcionam maior velocidade e qualidade. Entretanto, vale a pena você saber que além destes, foram desenvolvidos simultaneamente mais uma série de outros programas, abrangendo as áreas educacionais, vídeo-games, jogos inteligentes e profissionais, que podem ser adquiridos em lojas e revendedores autorizados CCE.

Na ordem de gravação da fita, temos o seguinte:

- 1- MUSIC COMPOSER (Compositor de Músicas)
É um programa que utiliza as características únicas do processador de som do seu MC-1000, permitindo a composição em três canais independentes.
- MARS LANDING (Aterrissagem em Marte)
Teste sua capacidade como piloto de uma nave espacial. Aterrisse em Marte com o mínimo gasto de combustível. Vença o desafio do universo.
- 3- CITY BOMBER (Bombardeio na Cidade)
Pilote com destreza o seu helicóptero e destrua a cidade inimiga, mas cuidado, pois ela poderá surpreendê-lo.
- 4- KINDERGARTEN MATH (Matemática para Jardim da Infância)
Quem melhor que um computador para ensinar os princípios da matemática? Veja aqui como brincando uma criança pode aprender a gostar de desenvolver sua capacidade.
- 5- TANK BATTLE (Batalha com Tanques)
Sua habilidade em dirigir e atirar com um tanque será colocada a prova contra o seu inimigo nesta batalha sensacional.

Agora, nas páginas seguintes, você encontrará as instruções para o carregamento e manuseio dos programas.

INSTRUÇÕES

- 1- Ajuste o volume do seu gravador para aproximadamente 2/3 do curso.
- 2- Digite TLOAD (ou pressione <SHIFT> <T>).
- 3- Pressione a tecla <RETURN> e em seguida a tecla PLAY do seu gravador. Enquanto o programa estiver carregando, a cor da tela mudará num intervalo de tempo de aproximadamente 3 segundos.
- 4- Quando carregado, o programa aparecerá automaticamente na tela.
- 5- Se quiser carregar outro programa, pressione <SHIFT> <RESET> e repita as instruções acima.
- 6- As teclas <CTRL> <H> tem a finalidade de alterar o nível dos programas.

MARS LANDING

- 1- Usa joystick e <CTRL> <H>.
- 2- Tela 1 - apenas para um jogador
Tela 2 - para dois jogadores
Tela 3 - para dois jogadores (mais difícil)
Tela 4 - placar geral

Identificação das telas

Ver na parte inferior da tela

Exemplo:

```
1 - Fuel  
2 - Fuel ----- <-  
3 - Fuel ----- ->
```

Obs.: na tela 2 começará com o jogador do joystick da ESQUERDA
na tela 3 começará com o jogador do joystick da DIREITA

- 3- Pressione as teclas <CTRL> <H> e selecione a tela desejada.
- 4- Com o joystick, dirija a sua espaçonave e aterrise nas bases espaciais.
- 5- Há três bases de aterrisagem: base X1 (1 ponto), X3 (3 pontos) e X8 (8 pontos).
- 6- O botão do joystick serve para disparar os retrofoguetes (isso gasta combustível!).
- 7- A faixa vermelha na parte inferior da tela, indica o nível do combustível. Quando o combustível acabar, o jogo estará terminado.
- 8- Na parte superior da tela (HI), ficará registrada a maior pontuação alcançada. Chegando aos 100 pontos, aparecerá uma espaçonave diferente e o jogo se tornará mais emocionante.

CITY BOMBER

- 1- Este jogo contém 4 níveis com 4 velocidades diferentes. É para apenas um jogador.
- 2- Pode ser jogado com joystick ou no próprio teclado do MC-1000.

JOYSTICK: para seleccionar o nível e a velocidade, veja a figura na tela, e mova a alavanca no sentido desejado. Aperte o botão para soltar uma bomba.

TECLADO: <Y> - nível e velocidade 1
<1> - nível e velocidade 2
<I> - nível e velocidade 3
<Q> - nível e velocidade 4
<9> - solta a bomba

- 3- Seu helicóptero voa mais baixo a cada passagem. Portanto, destrua os prédios ou o seu helicóptero baterá, e o jogo estará terminado.
- 4- Para interromper o jogo, pressione as teclas <CTRL> e <H>.

KINDERGARTEN-MATH

- 1- O nível de jogo aparecerá na parte superior esquerda da tela. Pressionar as teclas <CTRL> <H> seleciona o nível desejado.
- 2- Contém 5 níveis. Nos níveis 2 a 4, há três diferentes velocidades: 1, 2 e 3. Velocidade 1 - lento, 2 - médio e 3 - rápido.

Após selecionar o nível de jogo, pressione a tecla 1, 2 ou 3 do seu MC-1000 e o jogo iniciará.

- 3- Nível 1 - é apenas para obter noções de quantidades e números de 1 a 10.

Nível 2 - Você deverá teclar 1, 2 ou ... 10, conforme a quantidade de figuras. Se acertou, aparecerá uma figura em movimento circular e, se errou, em movimento X.

Nível 3 - pressione o botão do joystick e selecione um número existente nas figuras da direita. Puxando ou empurrando a alavanca, posicione em frente a figura de mesmo número e atire, movendo a alavanca para a direita, mas faça-o rapidamente para eliminar todas as figuras antes que o tempo se esgote. Eliminando todas as figuras 5 vezes, aparecerá na tela uma figura sorridente parabenizando-o. Para iniciar, pressione as teclas <CTRL> <H> e selecione a velocidade desejada (ver item 2).

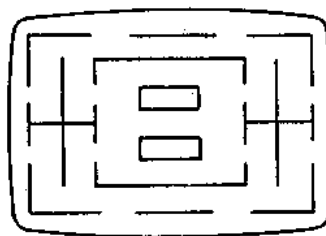
Nível 4 - Este é um jogo de memória. No quadro, aparecerá uma figura por alguns segundos, que você deverá memorizar. Em seguida, aparecerão várias figuras em volta. Com a alavanca do joystick, posicione a figura em movimento sobre a figura memorizada e aperte o botão. Se acertou, aparecerá uma figura em movimento circular e, se errou, em movimento X.

Nível 5 - Este é um jogo de memória. Durante alguns segundos serão mostradas 10 figuras para serem memorizadas. Em seguida, as figuras serão cobertas e, no topo da tela, aparecerá uma figura. Se esta for uma das figuras cobertas, puxe a alavanca do joystick em sua direção; se não, aperte o botão para selecionar a figura desejada. Descobertas as 10 figuras, aparecerá na tela uma figura sorridente parabenizando-o. Para reiniciar, pressione as teclas <CTRL> <H>.

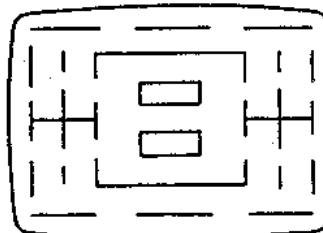
TANK BATTLE

- 1- Usa somente joysticks e teclas <CTRL> e <H>.
- 2- Este jogo contém 17 modos, e 4 diferentes telas de jogo. Pressionar <CTRL> <H> seleciona o nível desejado.
- 3- Níveis de 1 a 8 são para dois jogadores. A velocidade nos jogos 5 a 8 e 13 a 16 são quatro vezes mais rápidas. Jogos de 9 a 16 são jogados contra o computador.
- 4- As quatro diferentes telas são:

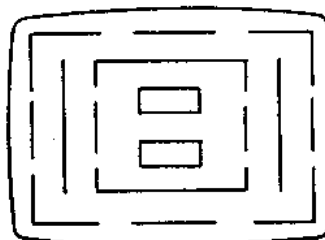
jogos 1,5,9,13



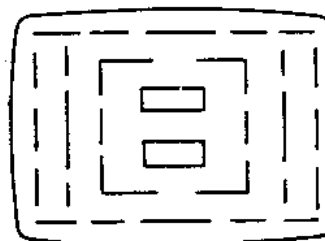
jogos 2,6,10,14



jogos 3,7,11,15



jogos 4,8,12,16



17 contém o placar geral.

MUSIC COMPOSER

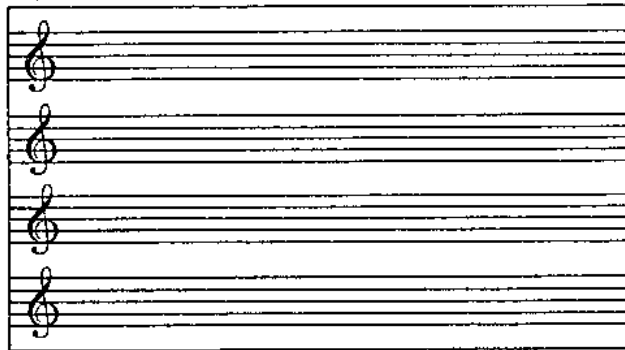
Agora você e toda a sua família podem tornar-se músicos de maneira divertida e fácil com seu MC-1000. Nenhum treino é necessário, basta seguir as instruções descritas nesse manual e breve você estará compondo sua própria música através do computador. É interessante que você tenha pelo menos noções básicas de música para facilitar a compreensão das explicações que daremos a seguir.

OS TRÊS CANAIS DA MÚSICA

Dividiremos a música em três partes, como nas partituras de órgão: melodia (mão direita), acompanhamento (mão esquerda) e ritmo ou contrabaixo (pedaleira).

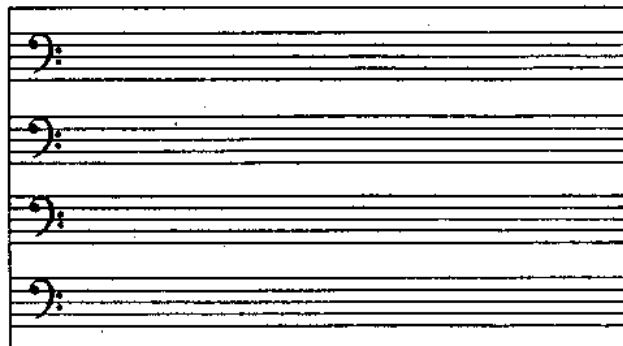
1- MELODIA

Na sua tela aparecerão quatro pautas (quatro grupos de 5 linhas horizontais) que você usará ao escrever a melodia. Note que as pautas estão em clave de SOL. Este, nós chamaremos "canal A".



2- ACOMPANHAMENTO

Na sua tela aparecerão quatro pautas que você usará para escrever o acompanhamento. Note que elas estão em clave de FA. Este, nós chamaremos de "canal B".



3- RITMO, OU CONTRABAIXO

Novamente, quatro pautas aparecerão na sua tela. Esta você usará para escrever o ritmo, e será chamado de "canal C".

Estes três canais são combinados quando você estiver ouvindo uma música própria ou uma das suas músicas favoritas. Entretanto, quando você estiver escrevendo, tratará dos canais individualmente.

Comece compondo no canal A e, depois componha o acompanhamento e o ritmo nos canais B e C respectivamente. Quando você quiser que seu computador toque a música, ele irá combinar os três canais.

PROCEDIMENTO PARA A COMPOSIÇÃO

- 1- Para programar seu MC-1000 para composição de música, coloque sua fita "Music Composer" no seu gravador e siga as instruções de carregamento de programa de fitas. Logo você ouvirá uma música agradável, e as notas da melodia (canal A) irão aparecendo na tela.


Se você esperar até que a música pare, seu computador selecionará automaticamente o canal A. Entretanto, não há necessidade de esperar: pressione <C>, e o computador imediatamente selecionará o canal A. Seu MC-1000 ficará aguardando que você comece a compor.

Se desejar ouvir a música novamente, pressione as teclas <CTRL> <H>. Pressionando novamente, a música será interrompida. Teclando <C> durante a música, seu computador selecionará a 4ª tela (tela em branco) onde você poderá ouvir o tom de cada tecla, aparecendo na tela as teclas correspondentes.

Para retornar ao canal A, pressione <CTRL> <H>. Quando a música iniciar, pressione novamente as teclas <CTRL> <H>.

2- Sua música consiste em notas agudas ou graves, curtas ou longas. Cada um dos três canais pode acomodar até 256 notas. As notas podem variar de um G grave, abaixo do C médio, até um F e G agudo, acima do C médio, como mostrado abaixo:

Quadro 1



NOTA { SOL LA SI DO RE MI FA SOL LA SI DO RE MI FA SOL
 G A B C D E F G A B C D E F G
 TECLA { SHIFT SHIFT
 8 6 7 1 2 3 4 5 6 7 1 2 3 4 5

Uma vez que você definiu a nota, você deve escolher sua duração (tempo durante o qual ela soará). As notas que você irá usar podem variar de uma semicolcheia (uma nota de duração muito curta) a uma semibreve (uma nota de longa duração), como indicado abaixo:

Quadro 2

	SEMI COLCHEIA		COLCHEIA		SEMININA		NINA		SEMI BREVE
TEMPO	1/16	3/32	1/8	3/16	1/4	3/8	1/2	3/4	1/1
TECLA	0	1	2	3	4	5	6	7	8
NOTAÇÃO DO MC-1000									
NOTAÇÃO MUSICAL PADRÃO									

Você deve ter percebido que as notas do MC-1000 que aparecem na tela de sua TV não são sempre idênticas, em aparência, àsquelas usadas na notação musical padrão. Não se assuste: use a tabela acima sempre que estiver compondo.

Ao compor no seu MC-1000, você deve primeiro digitar a nota e depois sua duração, antes de entrar com a próxima nota. Por exemplo, suponha que você entrou com o seguinte:

<1> <8> <3> <6> <5> <4> <1> <4> <2> <7> <3> <4> <2> <8>

Aparecerá na tela de sua TV o seguinte:



Note que você digitou um total de 14 números para estas 7 notas aparecerem na primeira pauta.

- 3- Depois que você terminou de entrar sua música, você pode ouvir o seu computador tocá-la.

Primeiro você deve decidir quantas vezes deseja ouvir a sua música. Para ouvir uma vez, tecle primeiro <A> e depois <E>. Para ouvir repeditamente, primeiro tecle e depois <E>. Pressionando <C>, a música será interrompida.

- 4- Se você terminou de escrever a melodia, agora é hora de começar a escrever o acompanhamento. Pressione a tecla <0>, e seu computador irá selecionar o canal B.

Para escrever o ritmo, pressione novamente a tecla <0> e o seu computador selecionará o canal C.

- 5- Agora que você terminou de escrever sua música, poderá ouvir todas as três partes (canais A, B e C) tocadas ao mesmo tempo. Seu MC-1000 pode tocar sua música em três velocidades diferentes: regular, rápido e super rápido.

Regular: O MC-1000 assume sempre esta velocidade, exceto quando você pede para tocar de outro modo.

Rápido: antes de pressionar <E>, você deve pressionar a tecla <G> e sua música será tocada em velocidade rápida.

Super rápido: se você pressionar <H> antes de pressionar <E>, a sua música será tocada ainda mais rápida.

Após ter ouvido a versão rápido e super rápido, se você preferir voltar a um tempo mais lento, pressione <CTRL> <H> e depois <E>. Assim sua música será tocada na velocidade regular.

Após você ter acabado de escrever um canal, poderá não estar completamente satisfeito com ele. Não há necessidade de recomeçar desde o princípio. É muito fácil fazer alterações no que você já escreveu:

1- TECLAS DE MUDANÇA DE POSIÇÃO

<T> : movimenta para a esquerda.

Cada vez que você pressionar esta tecla, o cursor irá mover uma nota para esquerda, em sua tela.

<Y> : movimenta para a direita.

Cada vez que você pressionar esta tecla, o cursor irá mover uma nota para a direita.

Posicione o cursor embaixo da nota que deseja alterar e redigite.

2- TECLAS PARA INCLUIR OU REMOVER NOTAS

<D> : apagar

Primeiramente use <T> ou <Y> afim de posicionar o cursor embaixo da nota que você deseja remover. Pressione <D> e a nota desaparecerá da sua tela.

Observação: ao pressionar somente as teclas <T> ou <Y>, você não afetará as notas já escritas na tela.

<I> : inserir

Primeiramente use <T> ou <Y> para mover o cursor até o lugar onde você queira acrescentar uma nota. Pressione <I> e as notas serão deslocadas para a direita, ficando apenas um espaço. Agora você poderá digitar a nota que quiser inserir.

Observação: se desejar acrescentar mais de uma nota, pressione <I> quantas vezes for necessário.

Se você quiser fazer alterações na sua música após ter mudado de canal ou tocado para ouvir, deverá seguir o procedimento descrito abaixo:

- a) primeiro, selecione o canal que deseja fazer alterações, pressionando a tecla <0>;
- b) pressione <Q>, e as notas daquele canal serão mostradas;
- c) Você pode então usar <T>, <Y>, <D> e/ou <I> para fazer qualquer alteração.

Como exemplo, vamos tentar escrever uma música popular chinesa de pescaria. Nós iremos passo a passo, através de todo o procedimento, com você.

Em primeiro lugar, ligue a sua TV, coloque a sua fita "MUSIC COMPOSER" no gravador e siga as instruções de "Procedimento para a Composição". O canal A está agora na tela. Comece a teclar as notas. Lembre-se: primeiramente tecle a nota e depois a duração, antes de teclar a próxima nota.

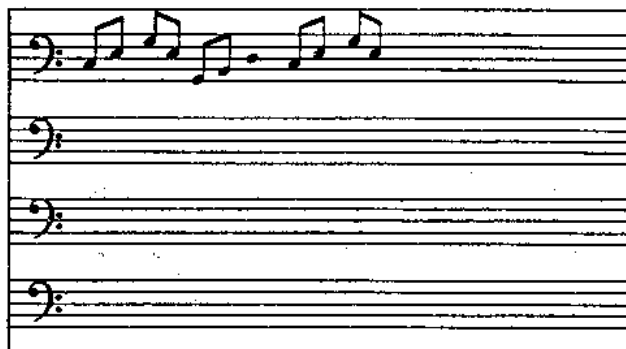


A primeira nota é DO: então você deve pressionar <1> (quadro 1). Como é uma semínima, pressione agora <4> (quadro 2). Continue fazendo isto para cada nota da música, seguindo a tabela abaixo:

1 ⁴	3 ⁴	2 ³	3 ⁰	2 ²	1 ²		5 ⁴	5 ⁴	5 ⁶
6 ⁴	5 ⁴	3 ³	5 ⁰	3 ²	2 ²		1 ⁴	1 ⁴	1 ⁶
1 ⁴	1 ⁴	6 ³	2 ²	5 ²		6 ⁴	5 ⁴	3 ⁵	2 ²
1 ⁴	5 ⁴	3 ³	2 ⁰	3 ²	2 ²		1 ⁴	1 ⁴	1 ⁶

Cada um dos números relacionados acima representa uma nota. Os números maiores representam o valor e os menores representam a duração. Por exemplo, a primeira nota desta música é representada como 1 (isto significa que é um DO, e sua duração é 1/4 (semínima)). Um ponto acima ou abaixo de um número significa que você deve pressionar <SHIFT> e o número.

Agora, para praticar, digite o seguinte acompanhamento no canal B (lembre-se de pressionar <0> primeiro para selecionar o canal B).



Quando você terminar de digitar todas as notas, pressione <A> e <E> para ouvir a música.

APÊNDICE A

MENSAGENS DE ERRO

Depois de ocorrer um erro, o computador exibirá uma mensagem de erro, pára a execução do programa e retorna para o modo direto.

Mensagens de erro são exibidas nos seguintes formatos:

```
Erro no modo direto      ?XX ERRO
Erro no modo indireto   ?XX ERRO EM LL
```

onde XX é o código do erro e LL é o número da linha onde o erro ocorreu.

Abaixo mostramos uma lista de códigos de erro.

<u>CÓD. ERRO</u>	<u>SIGNIFICADO</u>
II	INDICE ILEGAL
	Este erro ocorre se houver uma tentativa de acessar um elemento da matriz cujo índice está fora das dimensões da mesma.
NC	NÃO PODE CONTINUAR
	Tentativa de usar o comando CONT quando não existe programa na memória ou depois de um erro ter ocorrido.
MR	MATRIZ REDIMENSIONADA
	Foi feita uma tentativa de redimensionar uma matriz ou dimensioná-la após a mesma ter sido referenciada na execução (definida ou usada).
PI	PARAMETRO ILEGAL
	- Log com argumento negativo ou zero. - Um índice de matriz negativo. - SQR com argumento negativo. - Argumento impróprio usado nas funções MID\$, LEFT\$, RIGHT\$, IMP, OUT, WAIT, PEEK, POKE, TAB, SPC, STR\$, ou em instruções ON...GOTO, ON...GOSUB.
DI	DIRETO ILEGAL
	INPUT e DEF são inválidos no modo direto.

<u>COD. ERRO</u>	<u>SIGNIFICADO</u>
CL	STRING LONGA DEMAIS Tentativa de criar uma string maior que 255 caracteres.
FO	FALTA OPERANDO Nenhum operando após o operador.
NF	NEXT SEM FOR O loop FOR...NEXT não foi formado de maneira correta,. Nenhuma instrução FOR foi executada antes da instrução NEXT para a mesma variável.
FD	FIM DE DADOS Os dados não são suficientes para a instrução READ.
FM	FORA DA MEMÓRIA Matrizes ou programas muito grandes. Todo o espaço da memória está sendo usado.
FC	FORA DA STRING Variáveis tipo string ocupando mais espaço do que lhes foi reservado. Use strings menores ou menos variáveis.
RG	RETURN SEM GOSUB Uma instrução RETURN é encontrada antes do computador ter executado uma instrução GOSUB.
SN	ERRO DE SINTAXE Pontuação ou ortografia incorreta, caracter incorreto, etc.
CC	STRING COMPLEXA Dividir uma string muito complexa em duas ou mais strings.
TI	TIPO INCOMPATÍVEL Tentativa de relacionar um campo alfanumérico a uma variável numérica ou vice-versa.

<u>COD. ERRO</u>	<u>SIGNIFICADO</u>
FI	FUNÇÃO INDEFINIDA Referência feita a uma função que não foi previamente definida.
LI	LINHA INDEFINIDA Referência feita a um número de linha que não existe.
DZ	DIVISÃO POR ZERO Um número foi dividido por zero, ou zero foi elevado a uma potência negativa.
SE	SEM ESPAÇO Utilização de uma variável maior que 3.4E38.
IGNOROU EXTRA	 Numa instrução INPUT, foram fornecidos mais dados do que o número de variáveis definidas.
REENTRE DADOS	 Ocasionalmente dependendo da entrada que você fez para um INPUT e o que era esperado esta mensagem será exibida. Caso isto ocorra, simplesmente redigite os dados.
REBOBINE A FITA E REAPERTE O BOTAO PLAY	 Esta mensagem é exibida quando houver uma falha no gravador ou na fita. Quando ocorre este erro, o computador fica pronto para receber dados do gravador.

APENDICE B

PALAVRAS RESERVADAS

ABS	ON
AND	OR
ASC	OUT
ATN	PEEK
AUTO	PLOT
CALL	POKE
CHR\$	POS
CLEAR	PRINT
CONT	PR#
COS	READ
DATA	REM
DEBUG	RESTORE
DEF	RETURN
DIM	RIGHT\$
DRAW	RND
EDIT	RUN
END	SAVE
EXP	SAVE*
FAST	SET
FN	SGN
FOR	SIN
FRE	SLOW
GOSUB	SOUND
GOTO	SPC
GR	SQR
HOME	STOP
IF	STR\$
INP	TAB
INPUT	TAN
INT	TEMPO
INVERSE	TEXT
LEFT\$	THEN
LEN	TLOAD
LET	TO
LIST	USR
LOAD	UNDRAW
LOAD*	UNPLOT
LOG	VAL
MID\$	WAIT
NEW	WIDTH
NEXT	
NORMAL	
NOT	

APÊNDICE C

ACESSÓRIOS OPCIONAIS

O fabricante deste computador também fornece a você os seguintes acessórios opcionais que lhe serão de grande utilidade:

- monitores;
- drives;
- cartão de interface para drive;
- cartão de expansão;
- cartão de 80 colunas;
- software de jogos e aplicativos;
- interface para impressora.

Caso você não consiga encontrar algum destes itens, seu revendedor CCE poderá lhe indicar onde e quando o mesmo estará disponível para compra.

Sugestões para o aperfeiçoamento do manual, bem como indicações de possíveis falhas encontradas no decorrer deste são bem vindas.

Favor enviá-las à:

CCE - Indústria e Comércio de Componentes Eletrônicos S/A
DIVISÃO DE INFORMÁTICA
Av. Otaviano Alves de Lima, 2724
CEP 02501 - São Paulo - SP.

NOME : _____

PROFISSÃO : _____

CARGO : _____

TELEFONE : _____

MANUAL : _____

SUGESTÃO : _____



INFORMÁTICA

DATA CASSETE

e-books

Esta cópia tem como objetivo a divulgação e preservação de documentos que fazem parte da história da informática.